

AD-A072 775

BOLT BERANEK AND NEWMAN INC ARLINGTON VA
A COMPUTER PROGRAM FOR BEAM NOISE PREDICTION, (U)
MAY 79 R M ZESKIND, W L SCOTT
BBN-3654

F/6 9/2

N00173-77-C-0296

NL

UNCLASSIFIED

| OF |

AD
A072775





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Bolt Beranek and Newman Inc.



LEVEL II

Report No. 3654

12

Computer Program for
Noise Prediction

R.M. Zeskind, W.L. Scott



May 1979

Prepared for:
Naval Research Laboratory

DDC FILE COPY

This document has been approved
for public release and sale; its
distribution is unlimited.

79 05 30 217

Report No. 3654

6

A COMPUTER PROGRAM FOR BEAM NOISE PREDICTION,

10

R. M. / Zeskind
W. L. / Scott

11

May 9 1979



15

Contract No. N00173-77-C-0296
BBN Job No. 10601

12

88 p.

14

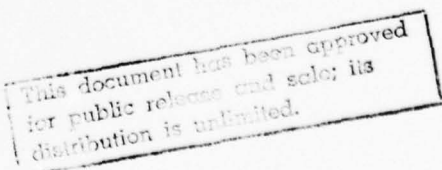
BBN-3654

Prepared for:

Dr. Orest Diachok
Code 8160
Naval Research Laboratory
Washington, D. C. 20375

Prepared by:

Bolt Beranek and Newman Inc.
1701 North Fort Myer Drive
Arlington, Virginia 22209



406 454

set

TABLE OF CONTENTS (continued)

	Page
4. PROGRAM DENS.....	4-1
4.1 Input Data File.....	4-1
4.2 Main Program.....	4-1
4.2.1 Common Areas.....	4-2
4.2.2 Description and Flowchart.....	4-2
4.3 Output.....	4-4
4.4 Glossary and Program Listing.....	4-5
4.5 Subroutines.....	4-11
4.5.1 Subroutine IFT.....	4-11
4.5.2 Subroutine POW2DB.....	4-14
4.5.3 Subroutine DIST.....	4-17

LIST OF FIGURES

	Page
Figure 1 Data Flow of Ambient Noise Prediction Computer Implementation.....	1-2
2 Flowchart for Program GEOM.....	2-4
3 A Polar Triangle.....	2-5
4 Route Segments and Triangles.....	2-7
5 Example of Terminal Printout from Program GEOM.....	2-9
6 Definition of Array Angles.....	3-4
7 Flowchart of Program CFUNC.....	3-7
8 Flowchart Function Z (R,D).....	3-33
9 Across-Route Density Function.....	3-36
10 Flowchart for Function BETA.....	3-38
11 Flowchart for Function INCLU.....	3-40
12 Subroutine GAMA Flowchart.....	3-43
13 Flowchart for SUBROUTINE BEAM.....	3-45
14 Program DENS Flowchart.....	4-3
15 Flowchart of Subroutine IFT.....	4-12
16 Flowchart of Subroutine POW2DB.....	4-16
17 Flowchart of Subroutine Dist.....	4-18

LIST OF TABLES

	Page
Table 1 Structure of Input File Datin.....	2-2
2 Structure of Output File GEO.....	2-10
3 Structure of Transmission Loss Files.....	3-2
4 Structure of Disk File Source.....	3-5
5 Structure of File PHIX.....	3-16
6 Structure of the Output of Program DENS.....	4-6

1. INTRODUCTION

The purpose of this report is to document the computer programs that have been written to implement the prediction algorithm described in BBN Report 3653 by M. Moll, R. M. Zeskind and W. L. Scott entitled "An Algorithm for Beam Noise Prediction". The authors assume that the reader is already familiar with the contents of that report.

1.1 Ambient Noise Prediction Computer Programs

The algorithms for calculating the probability density of the averaged noise power at the output of a beamformer in a specified passband resulting from ship traffic in an acoustic basin have been programed in FORTRAN for a CDC 660 digital computer. The algorithms are implemented as three separate programs and their associated data files. Figure 1 presents a block diagram of the data flow of the ambient noise prediction computer implementation.

The input information is divided into three types. Transmission loss data at a given sensor location is read into the computer and stored on a file as a table of transmission loss as a function of range and bearing. One such transmission loss file is created for each frequency of interest. Sensor, route and ship traffic information, which is independent of frequency, is read into the computer. This frequency-independent information is used as the input to a program which computes such quantities as route segment lengths, earth-centered angles between routes and sensor, and other geometric parameters needed in the computation of the characteristic function $\phi_y(\omega)$. The outputs of this geometry program are stored on a file for later use as inputs to the characteristic function program.

The characteristic function of beam noise power is computed via the algorithm derived in Section 2.0 of BBN Report 3653. This program reads in radiated noise data for each type of ship and other frequency-dependent input information such as beam pattern parameters. Beam steering angle and array orientation are also entered at this point. The characteristic function program reads in the data stored in the geometry file and the appropriate transmission loss file. The program calculates the mean and variance of received average noise power as well as its characteristic function and stores the results on a file to be used as input to the final computer program.

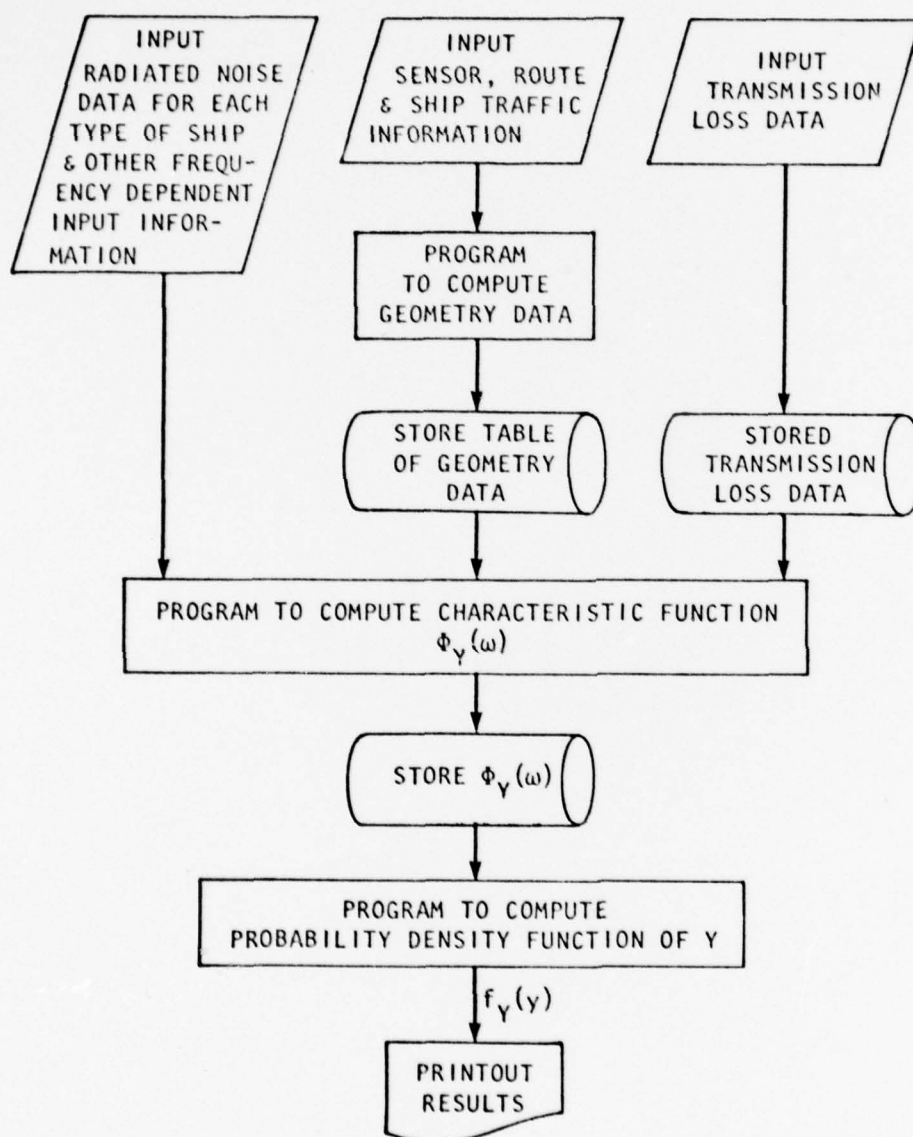


FIGURE 1 Data Flow of Ambient Noise Prediction Computer Implementation

The final program computes the probability density function of Y using the numerical algorithm described in Section 3.2 of BBN Report 3653. The mean, variance and probability density function for Y are printed out as the final result.

The structure of the program provides several advantages. The geometric data need be computed only once for a given sensor location and route structure. Since it is independent of frequency and beam steering angle, the same geometry data can be used many times by the characteristic function program. Furthermore, the transmission loss data need be entered only once for a particular sensor location.

1.2 Report Overview

The remainder of this report documents the individual computer programs shown in Figure 1. Section 2 describes Program GEOM which computes the geometry data. Also described are the structure of the input data file of sensor, route and ship traffic information; and the structure of the geometry data file. Section 3 describes Program CFUNK which computes the characteristic function $\Phi_Y(\omega)$. The structure of the input and output files needed for this program are also described in Section 3. Section 4 describes Program DENS, which computes the probability density function Y and prints out the results.

The documentation for each program consists of a brief description of the main program, the structure of input and output files, a glossary of variable names, flowcharts and listings of the program.

2. PROGRAM GEOM

In Section 2.2 of BBN Report 3653 were derived algebraic equations to be solved for certain geometric parameters needed in the computation of the characteristic function of received noise $\phi_Y(\omega)$. These geometric parameters depend only on the position of the sensor and the position of shipping routes on the earth's surface. They can be broken up into two categories. In the first category, the parameters depend on the variables of integration in the calculation of $\phi_Y(\omega)$. In the second category they do not. This memorandum documents a FORTRAN computer program called Program GEOM which computes the geometric parameters of the second category.

Program GEOM is structured to compute the geometric parameters for each individual segment of a shipping route. It is assumed that a route will have from one up to a maximum of five segments. The program assumes a maximum of ten routes. The program reads input data from a previously created disk file labelled DATIN and stores the results of the computations on a disk file labelled GEO. The results are also printed at the computer terminal so that a hard copy is available for documentation.

2.1 Input Data File

The input data for Program GEOM is read from file DATIN. This file contains the identification number for the particular case of sensor position and shipping routes to be processed; the sensor's position; the number of routes and the appropriate information for each route by segments. Table 1 gives the structure of the file DATIN. Each line of the table corresponds to one logical record on the file.

The following convention was assumed for the position of the sensor and of the route segment end-points. If the latitude is north, then the degrees, minutes and seconds that specify the latitude are all positive. If the latitude is south, then the degrees, minutes and seconds are entered as negative numbers. If the longitude is west, then enter the degrees, minutes and seconds as positive numbers. If the longitude is east, enter them all as negative numbers. It is also assumed that the segments are ordered from left to right along a route.

TABLE 1 STRUCTURE OF INPUT FILE DATIN
 (Note: Each Line Represent One Logical Record)

<u>DESCRIPTION</u>		<u>FORMAT</u>
Identification Number		(I3)
Sensor Latitude, Sensor Longitude (in degrees, minutes, seconds)		(6F6.1)
Number of Routes		(I3)
REPEATED FOR EACH ROUTE REPEATED FOR EACH SEGMENT OF A ROUTE	Number of Segments on This Route	(I3)
	Latitude and longitude of left-end-point of segment (in deg., min., sec.)	(6F6.1)
	Width of route segment at left end-point (in degrees)	(F8.4)
	Latitude and longitude of right end-point (in deg., min., sec.)	(6F6.1)
	Width of route at right end-point, maximum width of segment (both in degrees)	(2F8.4)
	Large, small and fishing ship densities on this segment (ships/n. mile)	(3F10.7)
	.	
.		
.		
etc.		

2.2 Program Description

Figure 2 is a flowchart for Program GEOM. The program first initializes constants and arrays; defines a function RADS to convert angles in degrees, minutes and seconds to radians; and rewinds the input and output files. Next, the identification number is read from files DATIN and also the sensor position. The sensor latitude and longitude are converted to radians. The angle between the north pole and sensor is computed as:

$$c_o = \frac{\pi}{2} - \text{latitude of sensor}$$

The angle between the Greenwich meridian and the sensor meridian is computed as:

$$W_o = \begin{cases} \text{sensor longitude, if positive} \\ 2\pi + \text{longitude, if negative} \end{cases}$$

The program next reads from file DATIN the number of routes and enters a DO LOOP to process information one route at a time. The first statement in the DO LOOP for the routes reads in the number of segments on this particular route and then enters a DO LOOP on the segments of this route. In the DO LOOP on the segments, the program reads in the data for each segment of the route and computes the geometric parameters for that segment. The segments are ordered from left to right along the route. The geometric parameters and shipping density information are stored in a three dimensional array G. The elements of array G are defined in the glossary below. The computations performed in this part of the program are as follows.

First the angles of a north polar triangle formed by the sensor, route end-point and north pole are computed. Figure 3 shows this triangle for a general end-point R_i . The side c_i is computed from;

$$c_i = \frac{\pi}{2} - \text{latitude of end-point}$$

The angle N_i is given by

$$N_i = |W_i - W_o|$$

where

$$W_i = \begin{cases} \text{longitude of end-point if positive} \\ 2\pi + \text{longitude of end-point if negative} \end{cases}$$

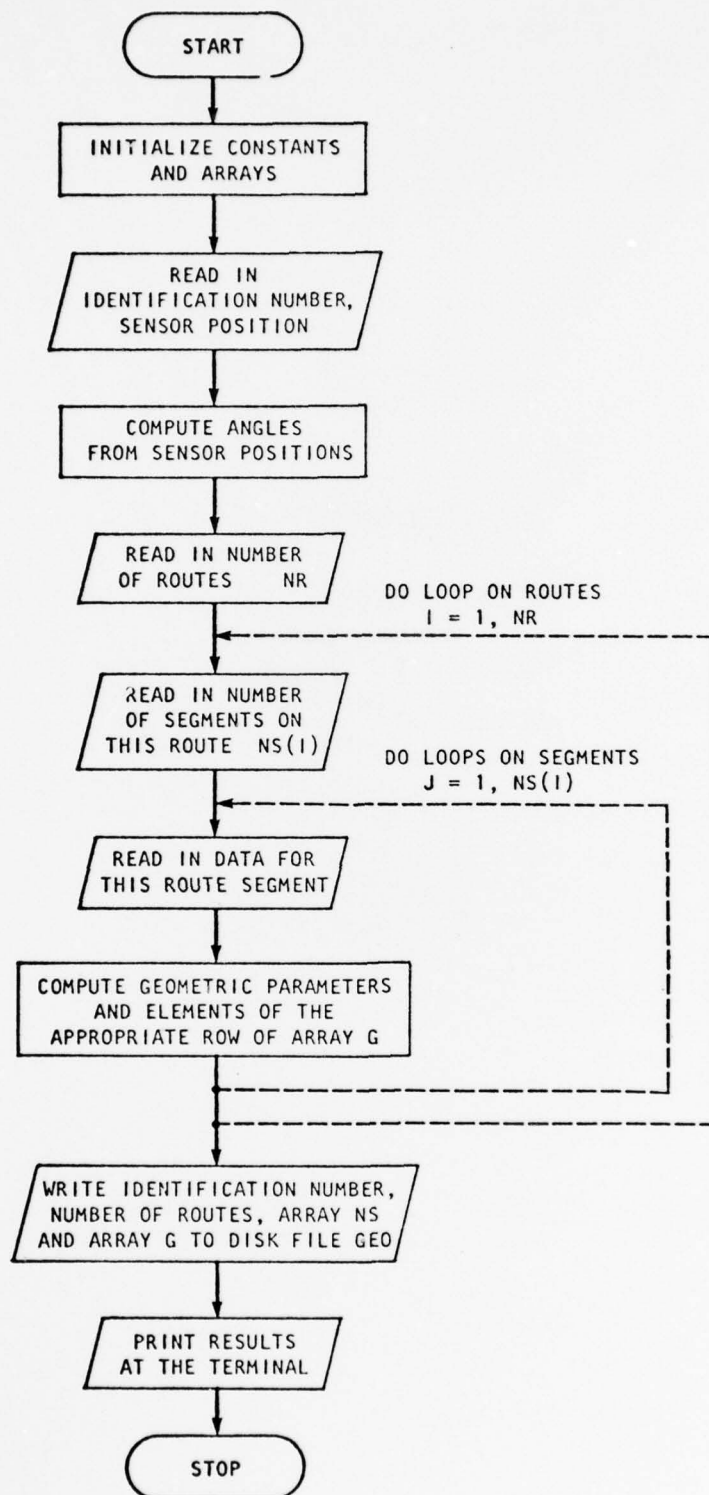
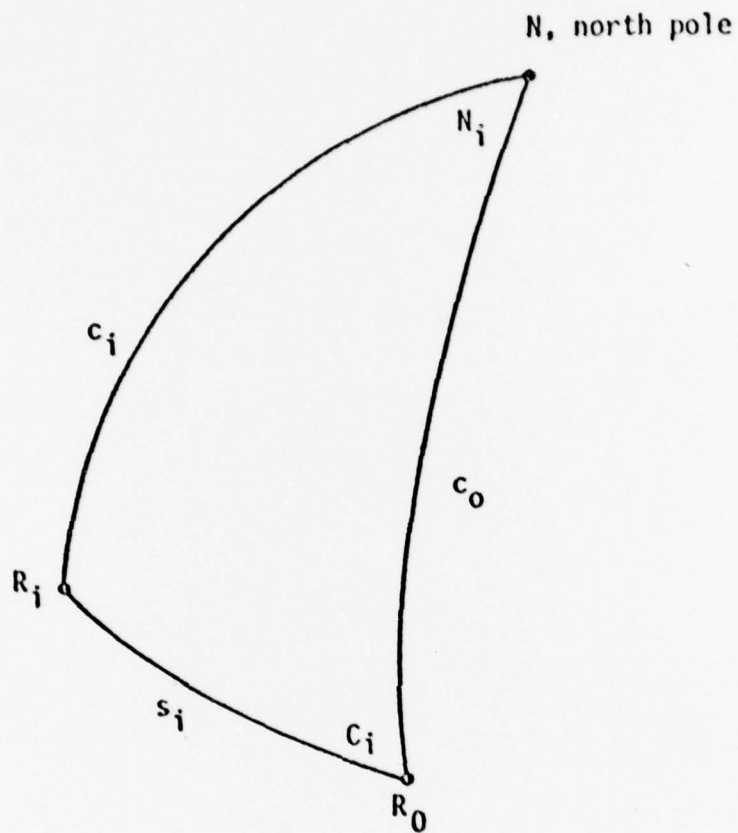


Figure 2 Flowchart for Program GEOM



R_0 = Sensor

R_1 = end-point of a route segment

Figure 3 A Polar Triangle

The side s_1 of the polar triangle is found from

$$\cos s_1 = \cos c_1 \cos c_0 + \sin c_1 \sin c_0 \cos N_1$$

and then taking the arc cosine. Next $\sin s_1$ and $\tan \frac{1}{2} s_1$ are computed. The angle C_1 of the polar triangle is found from the arc sine of

$$\sin C_1 = \sin c_1 \sin N_1 \div \sin s_1$$

The azimuth of point R_1 is computed as

$$Z_1 = C_1, W_1 \leq W_0$$

$$= 2\pi - C_1, W_1 > W_0$$

The triangles formed by the route segment end-points and the sensor are shown in Figure 4. The internal angles of the route segment triangles are computed as

$$I_1 = |Z_{i+1} - Z_i|$$

The side l_1 of the route segment triangle is found from taking the arc cosine of

$$\cos l_1 = \cos s_i \cos s_{i+1} + \sin s_i \sin s_{i+1} \cos I_1$$

Then the length of the route segment in nautical miles is computed from

$$L_1 = \rho l_1$$

where ρ is the radius of the earth in nautical miles.

The internal angle F_1 of the route segment triangle is computed from

$$\sin F_1 = \sin s_{i+1} \sin I_1 \div \sin l_1$$

and then taking the arc sine.

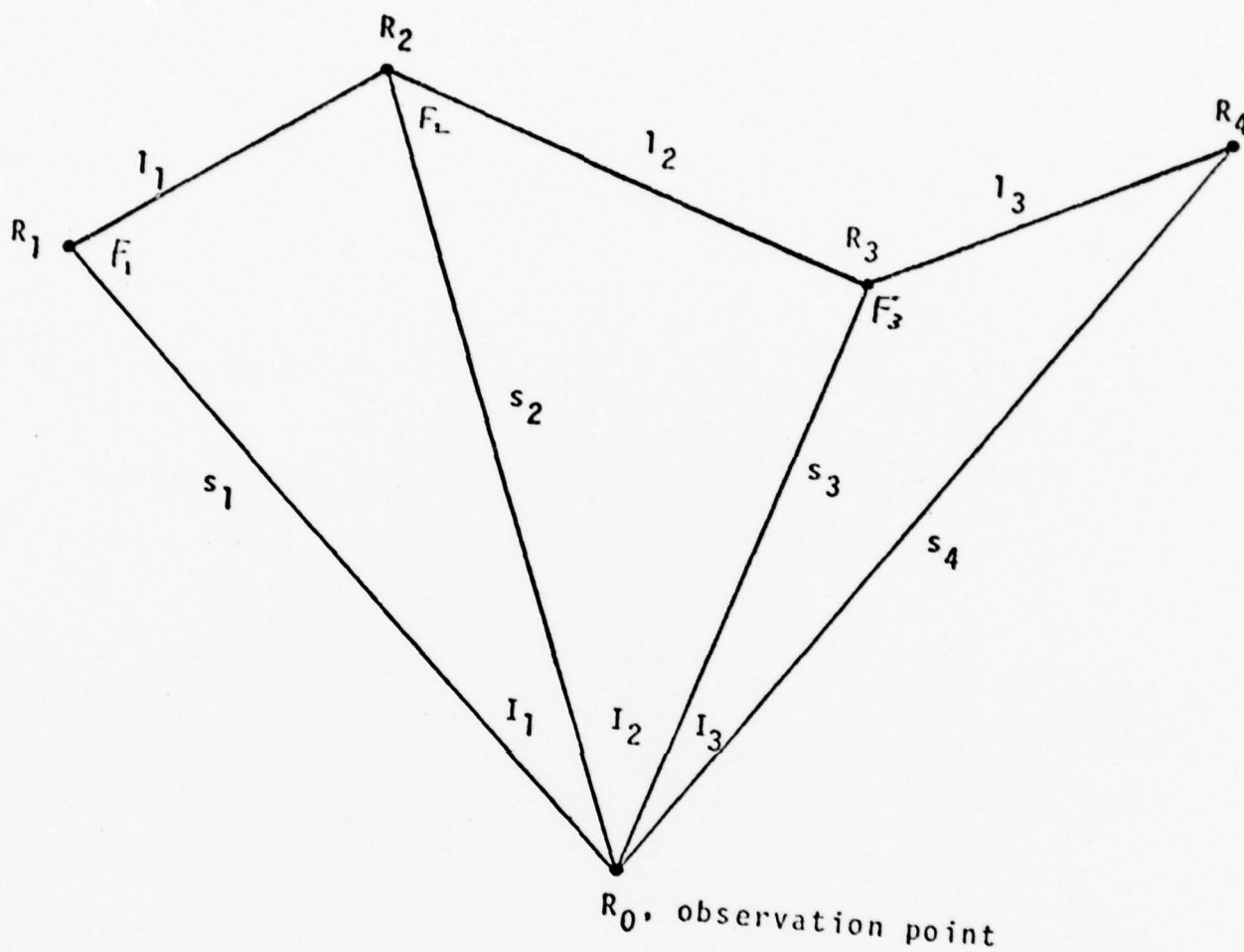


Figure 4 Route Segments and Triangles

The last computations in the DO LOOP on the segments is to compute the two route width coefficients b_i and c_i for this segment:

$$b_i = 2 \ell_i^{-1} \left[m_i - w_i + \sqrt{(m_i - w_i)(m_i - w_{i+1})} \right]$$

$$c_i = \ell_i^{-2} \left[(m_i - w_i) + (m_i - w_{i+1}) + \sqrt{(m_i - w_i)(m_i - w_{i+1})} \right]$$

where w_i is the route width at R_i

m_i is the maximum width of the segment $R_i R_{i+1}$

After the computations for each segment of a route have been completed, the program returns to the outer DO LOOP to process another route. Once all routes have been processed, the program enters the section of code which writes the results to the output file GEO. The structure of the output file GEO will be discussed below.

In the final section of Program GEOM, the results of the calculations are printed at the terminal. Figure 5 shows an example of some typical printout at the terminal. The first thing printed is a header which includes the identification number and the number of routes. Then the route and segment are identified and the thirteen parameters for that route segment are printed. This corresponds to one row of the array G.

2.3 Output Data File GEO

The output from Program GEOM to be used in further computations is stored on disk file GEO. File GEO is one of the input files for the program that computes the characteristic function of received noise, Program CPUNK. This file contains the identification number, number of routes, an array NS containing the number of segments on each route, and the array G of geometric parameters and ship densities. Table 2 gives the structure of file GEO.

OUTPUT FROM PROGRAM GEOM IDNUM= 1

NUMBER OF ROUTES = 10

ROUTE 1 SEGMENT 1

WIDTH= .4363E-02 B= .4466E+00 E= .5715E+00

SIN S= .8389E+00 COS S= .5442E+00 TAN 1/2S= .5433E+00

F= .3528E+00 Z= .5357E+01 I= .4371E+00 L= .2032E+04

SHIP DENSITIES

LARGE = 0.

SMALL = 0.

FISH = 0.

ROUTE 1 SEGMENT 2

WIDTH= .2182E-02 B= .6140E+00 E= .1045E+01

SIN S= .4548E+00 COS S= .8906E+00 TAN 1/2S= .2406E+00

F= .6910E+00 Z= .5794E+01 I= .5179E+01 L= .1527E+04

SHIP DENSITIES

LARGE = 0.

SMALL = 0.

FISH = 0.

ROUTE 2 SEGMENT 1

WIDTH= 0. B= .8606E+00 E= .2716E+01

SIN S= .8856E+00 COS S= .4645E+00 TAN 1/2S= .6047E+00

F= .4415E+00 Z= .5173E+01 I= .1029E+00 L= .6598E+03

SHIP DENSITIES

LARGE = 0.

SMALL = 0.

FISH = 0.

ROUTE 2 SEGMENT 2

WIDTH= .3491E-01 B= .1886E+00 E= .3913E+00

SIN S= .7934E+00 COS S= .6087E+00 TAN 1/2S= .4932E+00

F= .4419E-01 Z= .5276E+01 I= .5351E-01 L= .1738E+04

Figure 5 Example of Terminal Printout from Program GEOM

TABLE 2

STRUCTURE OF OUTPUT FILE GEO

(Note: Each Line Represents One Logical Record)

DESCRIPTION	FORMAT
Identification Number (IDNUM), number of routes (NR).	(1X, I3, 2X, I3)
Array of Number of Route Segments on Each Route [NS(J), where NS(1) = number of segments on route 1, etc.]	(1X, 10I3)
Array of Computed Geometric Parameters and Ship Densities. [G(I, J, K) is written one segment (ROW) in four logical record by routes]	(1X, 4E20.4)
.	.
.	.
.	.
(Until all segments on all routes are written)	.

2.4 Glossary and Program Listing

The following glossary contains definitions of the FORTRAN variable names used in Program GEOM. The names are presented in alphabetical order.

- AN = polar angle between sensor and route segment end-point measured from north pole (in radians) N_1
- C = side of the north polar triangle formed by north pole and route segment end-point (in radians) c_1
- CO = side of north polar triangle formed by north pole and sensor location (in radians) c_0
- D1 = degrees of latitude of segment end-point
- D2 = degrees of longitude of segment end-point
- G(I,J,K) = an array dimensioned (5x13x10) containing the computed geometric parameters and ship densities for each segment of each route. The index K indicates the route while the index I indicates the segment. For a given segment on a given route the elements in the I^{th} row are:
- G(I,1,K) = route segment left end-point width (radians) W_1
- G(I,2,K) = b_1 , a parameter used to model the route segment width envelope.
- G(I,3,K) = e_1 , a parameter used in the model of the route segment width envelope.
- G(I,4,K) = $\sin s_1$, sine of the earth centered angle s_1 between the sensor and a segment left end-point.
- G(I,5,K) = $\cos s_1$
- G(I,6,K) = $\tan \frac{1}{2} s_1$
- G(I,7,K) = F_1 an interior angle of the triangle formed by the segment and the sensor (in radians)
- G(I,8,K) = Z_1 the azimuth of the left end-point of the segment (in radians)

$G(I,9,K)$ = I_1 , an interior angle of the triangle formed by the segment and the sensor (in radians)
 $G(I,10,K)$ = length of the segment in nautical miles L_i
 $G(I,11,K)$ = density of large merchant ships on this segment (ships/n. mile)
 $G(I,12,K)$ = density of small merchant ships on this segment (ships/n. mile)
 $G(I,13,K)$ = density of fishing vessels on this segment (ships/n. mile)
IDNUM = an identification number for the particular sensor/route geometry
NR = number of routes (up to ten)
NS = an array dimensioned 10, each element of this array contains the number segments on a route.
PI = π
RHO = radius of the earth in nautical miles
SLA = an array dimensioned 3 which contains the sensor latitude as:
 SLA (1) = degrees
 SLA (2) = minutes
 SLA (3) = seconds
SLO = an array dimensioned 3 which contains the sensor latitude as:
 SLO (1) = degrees
 SLO (2) = minutes
 SLO (3) = seconds
S1 = seconds of latitude of segment end-point
S2 = seconds of longitude of segment end-point
TEMP
TEMP 2
TEMP 3
T1
T2
T3
T4
W = longitude of end-point in radians

} TEMPORARY STORAGE LOCATIONS

WIT = width of route segment at left end-point W_1
 WO = longitude of sensor in radians
 X1 = minutes of latitude of segment end-point
 X2 = minutes of longitude of segment end-point
 Z = temporary storage location

The listing of the program is as follows.

```

      PROGRAM GEOM(OUTPUT,DATIN,GEO,TAPE2=DATIN,TAPE6=GEO)
C PROGRAM TO COMPUTE THE GEOMETRIC COEFFICIENTS NEEDED IN
C AMBIENT NOISE MODEL. WRITTEN 15 FEB. 78 BY ZESKIND.
C DATIN = FILE OF INPUT SENSOR AND ROUTE DATA.
C GEO = FILE OF GEOMETRIC PARAMETERS TO BE USED AS
C INPUT TO KORE AND LINE COUNT PROGRAMS.
C
      DIMENSION G(5,13,10),NS(10),SLA(3),SLO(3)
      DATA G/650*0.0/,NS/10*0/
      RADS(D,XM,S)=(1.745329252E-02)*(D+(XM/60.)+(S/3600.))
      RHO=3437.746771
      PI=3.141592654
      REWIND 2
      REWIND 6
C READ IN IDENTIFICATION NUMBER
      READ(2,15) IDNUM
C READ SENSOR POSITION
      READ(2,10)(SLA(K),K=1,3),(SLO(J),J=1,3)
10  FORMAT(6F6.1)
      CO=RADS(SLA(1),SLA(2),SLA(3))
      CO=(PI/2.)-CO
      WO=RADS(SLO(1),SLO(2),SLO(3))
      IF(WO.LT.0.0) WO=(2.*PI)+WO
      T1=COS(CO)
      T2=SIN(CO)
C READ IN NUMBER OF ROUTES
      READ(2,15) NR
15  FORMAT(I3)
C ***** DO LOOP ON ROUTES *****
      DO 1010 K=1,NR
C READ IN NUMBER OF SEGMENTS ON THIS ROUTE
      READ(2,15) NS(K)
      NUMS=NS(K)
C ---- DO LOOP ON SEGMENTS ----
      DO 1000 J=1,NUMS

```

```

C READ IN SEGMENT DATA FOR LEFT END-POINT
  READ(2,10) D1,X1,S1,D2,X2,S2
  C=(PI/2.)-RADS(D1,X1,S1)
  W=RADS(D2,X2,S2)
  IF(W.LT.0.0) W=(2.*PI)+W
  AN=ABS(W-WO)
  T3=COS(C)
  T4=SIN(C)
  G(J,5,K)=T3*T1+T4*T2*COS(AN)
  TEMP=ACOS(G(J,5,K))
  G(J,4,K)=SIN(TEMP)
  IF(G(J,5,K).EQ.-1.0) GO TO 5
  G(J,6,K)=G(J,4,K)/(1.+G(J,5,K))
  GO TO 6
5  G(J,6,K)=1.0E+99
6  TEMP=T4*SIN(AN)/G(J,4,K)
  G(J,8,K)=ASIN(TEMP)
  IF(G(J,8,K).LT.0.0) G(J,8,K)=G(J,8,K)+(PI/2.)
  IF(W.GT.WO) G(J,8,K)=(2.*PI)-G(J,8,K)
C READ IN LEFT END-POINT ROUTE WIDTH
  READ(2,16) WIT
  G(J,1,K)=RADS(WIT,0.0,0.0)
C READ IN SEGMENT RIGHT END-POINT POSITION
  READ(2,10) D1,X1,S1,D2,X2,S2
  C=(PI/2.)-RADS(D1,X1,S1)
  W=RADS(D2,X2,S2)
  IF(W.LT.0.0) W=(2.*PI)+W
  AN=ABS(W-WO)
  T3=COS(C)
  T4=SIN(C)
  TEMP=T3*T1+T4*T2*COS(AN)
  TEMP2=ACOS(TEMP)
  TEMP3=SIN(TEMP2)
  TEMP2=T4*SIN(AN)/TEMP3
  Z=ASIN(TEMP2)
  IF(Z.LT.0.0) Z=Z+(PI/2.)
  IF(W.GT.WO) Z=(2.*PI)-Z
C COMPUTE ANGLE I
  G(J,9,K)=ABS(Z-G(J,8,K))
C COMPUTE LENGTH OF SEGMENT L
  TEMP2=G(J,5,K)*TEMP+G(J,4,K)*TEMP3*COS(G(J,9,K))
  TEMP=ACOS(TEMP2)
  G(J,10,K)=TEMP*RHO
C COMPUTE ANGLE F
  TEMP2=TEMP3*SIN(G(J,9,K))/SIN(TEMP)
  G(J,7,K)=ASIN(TEMP2)
  IF(G(J,7,K).LT.0.0) G(J,7,K)=G(J,7,K)+(PI/2.)
C READ IN RIGHT END-POINT WIDTH AND MAX WIDTH OF SEGMENT
  READ(2,16) WID,WMAX
16  FORMAT(2F8.4)
  WID=RADS(WID,0.,0.)

```

```

      WMAX=RADS(WMAX,0.,0.)
C  COMPUTE ROUTE WIDTH COEFFICIENTS FOR THIS SEGMENT
      T3=WMAX-G(J,1,K)
      T4=WMAX-WID
      G(J,2,K)=(2./TEMP)*(T3+SQRT(T3*T4))
      G(J,3,K)=(T3+T4+SQRT(T3*T4))/(TEMP*TEMP)
C  READ IN SHIPPING DENSITIES FOR SEGMENT
      READ(2,20)(G(J,L,K),L=11,13)
20    FORMAT(3F10.7)
1000  CONTINUE
1010  CONTINUE
C  **** OUTPUT RESULTS TO FILE GEO  ****
      WRITE(6,100) IDNUM,NR
100    FORMAT(1X,I3,2X,I3)
      WRITE(6,110) NS
110    FORMAT(1X,10I3)
      DO 115 K=1,NR
      NUMS=NS(K)
      DO 118 I=1,NUMS
      WRITE(6,120)(G(I,J,K),J=1,13)
120    FORMAT(1X,4E20.14)
118    CONTINUE
115    CONTINUE
C  ---- PRINT RESULTS AT THE TERMINAL ----
      PRINT 200,IDNUM
200    FORMAT(1X///1X,24HOUTPUT FROM PROGRAM GEOM,5X,7HIDNUM= ,I3)
      PRINT 210,NR
210    FORMAT(1X/1X,19HNUMBER OF ROUTES = ,I3///)
      DO 2010 K=1,NR
      NUMS=NS(K)
      DO 2000 N=1,NUMS
      PRINT 220,K,N
220    FORMAT(1X/1X,6HROUTE ,I3,5X,8HSEGMENT ,I3)
      PRINT 230,G(N,1,K),G(N,2,K),G(N,3,K)
230    FORMAT(1X//1X,7HWIDTH= ,E10.4,3X,3HB= ,E10.4,3X,
+ 3HE= ,E10.4)
      PRINT 240,(G(N,L,K),L=4,6)
240    FORMAT(1X//1X,7HSIN S= ,E10.4,3X,7HCOS S= ,E10.4,3X,
+ 10HTAN 1/2S= ,E10.4)
      PRINT 250,(G(N,M,K),M=7,10)
250    FORMAT(1X//1X,3HF= ,E10.4,3X,3HZ= ,E10.4,3X,
+ 3HI= ,E10.4,3X,3HL= ,E10.4)
      PRINT 260,(G(N,J,K),J=11,13)
260    FORMAT(1X//1X,14HSHIP DENSITIES/5X,8HLCARGE = ,E10.4,
+ 3X,8HSMALL = ,E10.4,3X,7HFISH = ,E10.4)
2000  CONTINUE
2010  CONTINUE
C
      REWIND 2
      REWIND 6
      STOP
      END

```

3. PROGRAM CFUNK

The algorithm for the computation of the characteristic function of beam noise power is described in Section 2.0 of BBN Report 3653. A computer program called CFUNK was written to implement this algorithm. Program CFUNK computes the mean, variance, and the real part of the characteristic function of beam noise power. These results, along with other pertinent information, is stored in a file, called PHIX, for use by Program DENS. Program DENS computes the probability density function. It is described in Section 4 of this report.

3.1 Input Files

Program CFUNK reads input data from three different files, as shown in Figure 1. These three input files for Program CFUNK are the file of geometry data called GEO, a transmission loss data file for the frequency of interest (TL10H for 10 Hertz) and a file of frequency dependent input information called SOURCE.

3.1.1 Geometry File

File GEO is the output file created by Program GEOM. Its structure was described in Section 2.3 above, and given in Table 2.

3.1.2 Transmission Loss File

Transmission loss data at a given sensor location is read into the computer and stored in a file as a table of transmission loss in power as a function of range and bearing. For this computer implementation it has been assumed that transmission loss is given at 10 nautical mile increments, starting at 10 nautical miles from the sensor, up to a maximum of 3500 nautical miles. It is also assumed that transmission loss data is available at five different azimuth angles. Thus the transmission loss data is stored in an array dimensioned 351 by 5, where the rows are range increments and the columns azimuthal angle. The first row gives the azimuthal angle in degrees.

The transmission loss files are labeled by frequency. For example, transmission loss at 10 Hertz is stored in a file labeled TL10H. Data at 50 Hertz would be stored in a file labeled TL50H, etc. The file name in the program statement at the beginning of the main computer program must be changed when a different frequency is to be used. Table 3 gives the structure of the transmission loss files.

Table 3 STRUCTURE OF TRANSMISSION LOSS FILES

(Note: Each line represents one logical record of input.)

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>
T1, T2, T3	Header for identifying file of the form TRANSMISSION LOSS 10 Hz	(1X, 25H)
TL(1,1), TL(1,2), ..., TL(1,5) . . .	Array of transmission loss data dimensioned (351 x 5) one row per logical record. First row contains azimuthal angles.	(1X, 5E15.8)
TL(351,1), TL(351,2), ..., TL(351,5)		

3.1.3 File SOURCE

The remaining input information that is necessary for program CFUNK is read in from a file called SOURCE. This file contains frequency-dependent input information and beam pattern parameters.

Table 4 gives the structure of input data file SOURCE. The file contains the mean and variance of the total radiated noise for large merchant, small merchant and fishing vessels at the frequency of interest. This is stored in a 3 by 2 dimensional array labeled S. The center frequency and frequency band in Hz are the next logical record on the file. The last logical record contains the beam pattern parameters. They are the azimuth of array broadside, BAZ; beam steering angle, BSTER; a beam pattern constant depending on frequency, ALPHA; and the main lobe beam width, BWIDE. All angles are in degrees. Figure 6 shows the definition of the angles associated with the array pattern.

3.2 Main Program

The main program of CFUNK performs the following functions: 1) initializing arrays and set program constants; 2) reads in input data; 3) computes initial parameters for numerical integration; 4) computes the probability of no noise; 5) computes the mean and variance of the averaged noise power at the output of the beamformer; 6) computes the real part of the characteristic function of averaged noise power; and 7) writes the results to the output file PHIX.

The following subprograms are called by the main program:

- Z(R,D) - returns a value of transmission loss for a given range and angle.
- BETA (Q, QO) - returns a value of the probability density function of transverse ship position across a route segment.
- BEAM - a subroutine to compute the normalized beam pattern for a given angle.
- GAMA - a subroutine to compute the real and imaginary parts of the characteristic function of the gamma probability density function.
- INCLU - a utility function subprogram which returns a value of one if an angle C is included between two angles A and B.

These subprograms are described in Section 3.5.

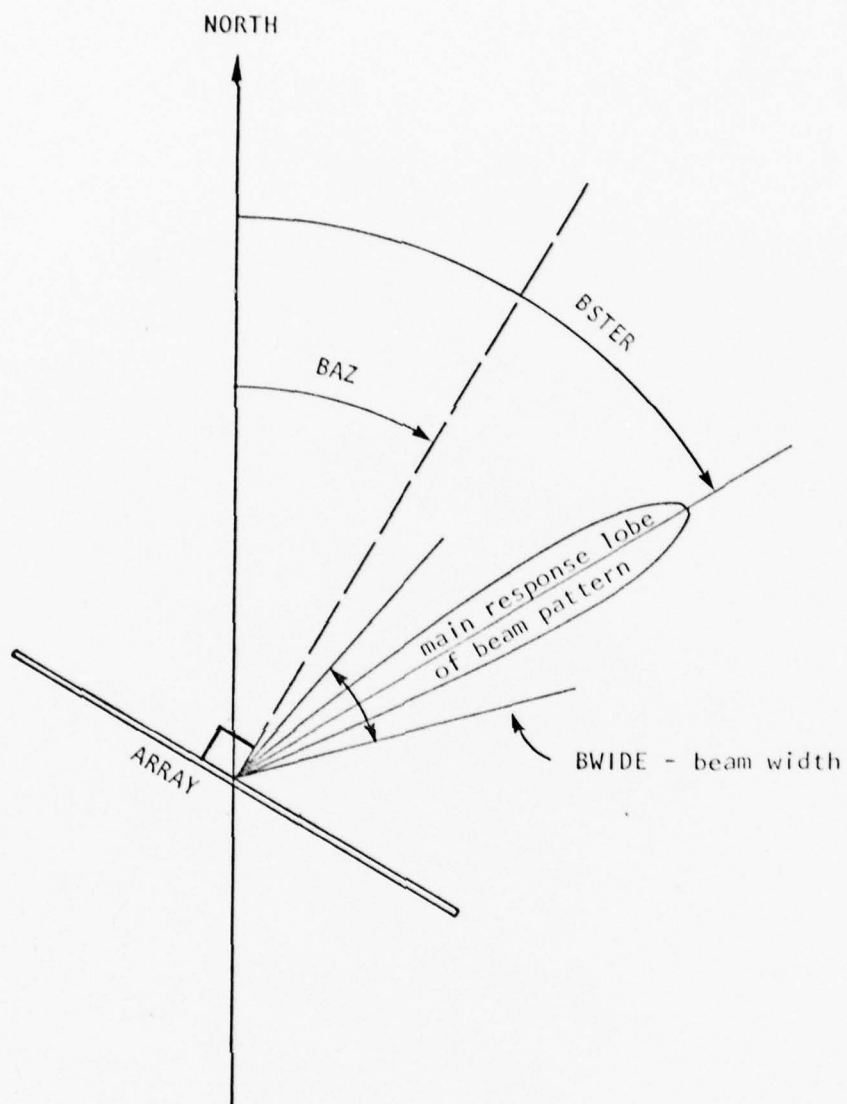


FIGURE 6 Definition of Array Angles

Table 4 STRUCTURE OF DISK FILE SOURCE

(Note: Each line represents one logical record of input.)

<u>Variable Name</u>	<u>Description</u>	<u>Format</u>
S(1,1), S(1,2)	mean, variance of large merchant ship radiated noise (pressure squared)	(2E20.13)
S(2,1), S(2,2)	mean, variance of small merchant ship radiated noise (pressure squared)	(2E20.13)
S(3,1), S(3,2)	mean, variance of fishing vessels radiated noise (pressure squared)	(2E20.13)
FREQ, DELF	center frequency, frequency band of noise (Hz)	(2E20.13)
BAZ, BSTER, ALPHA, BWIDE	azimuth of array broadside (in degr.), beam steering angle from north (in degr.), beam pattern constant, and main lobe beam width.	(4E18.10)

3.2.1 Common Storage

There is one block of common storage area labeled XLOSS. XLOSS contains the array TL of transmission loss data and is common to the main program and the function subprogram Z.

3.2.2 Flowchart and Description

Figure 7 presents the flowchart of the main program. At the start of the program, the arrays G, NS and PHI are initialized to zero. Program constants are set as follows:

RHO = 3437.746771 (radius of the earth in
nautical miles)

PI = 3.141492654.

TPI = 2π

DELR = 0.0029088821 radians. (range increment
step size equivalent to 10
nautical miles)

D10 = 0.1745329252 radians. (equivalent to 10°)

The following statement functions are defined at the start of the program:

- RADS - converts angles in degrees, minutes and seconds to radians.
- WIDTH - computes an approximation to route segment width
- XJ - computes the value of the Jacobian of the transformation between sensor coordinates and route coordinates.

The program next reads in the file GEO containing the geometric parameters computed in Program GEOM and stores them in array G. Next, the appropriate transmission loss data is read from file and stored in array TL. The first row of TL is converted from degrees to radians. Finally, the file SOURCE is read. This completes the input section of the program.

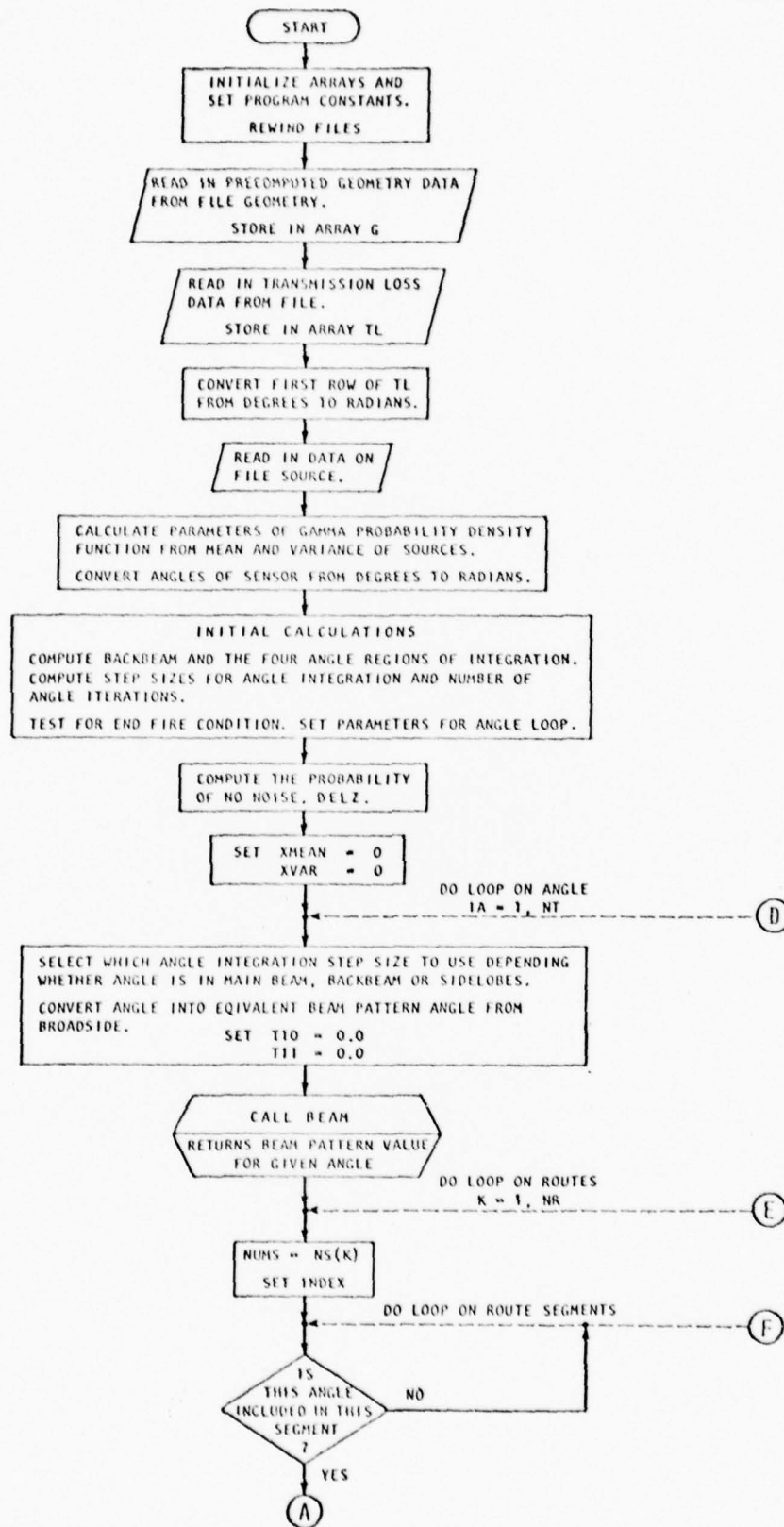
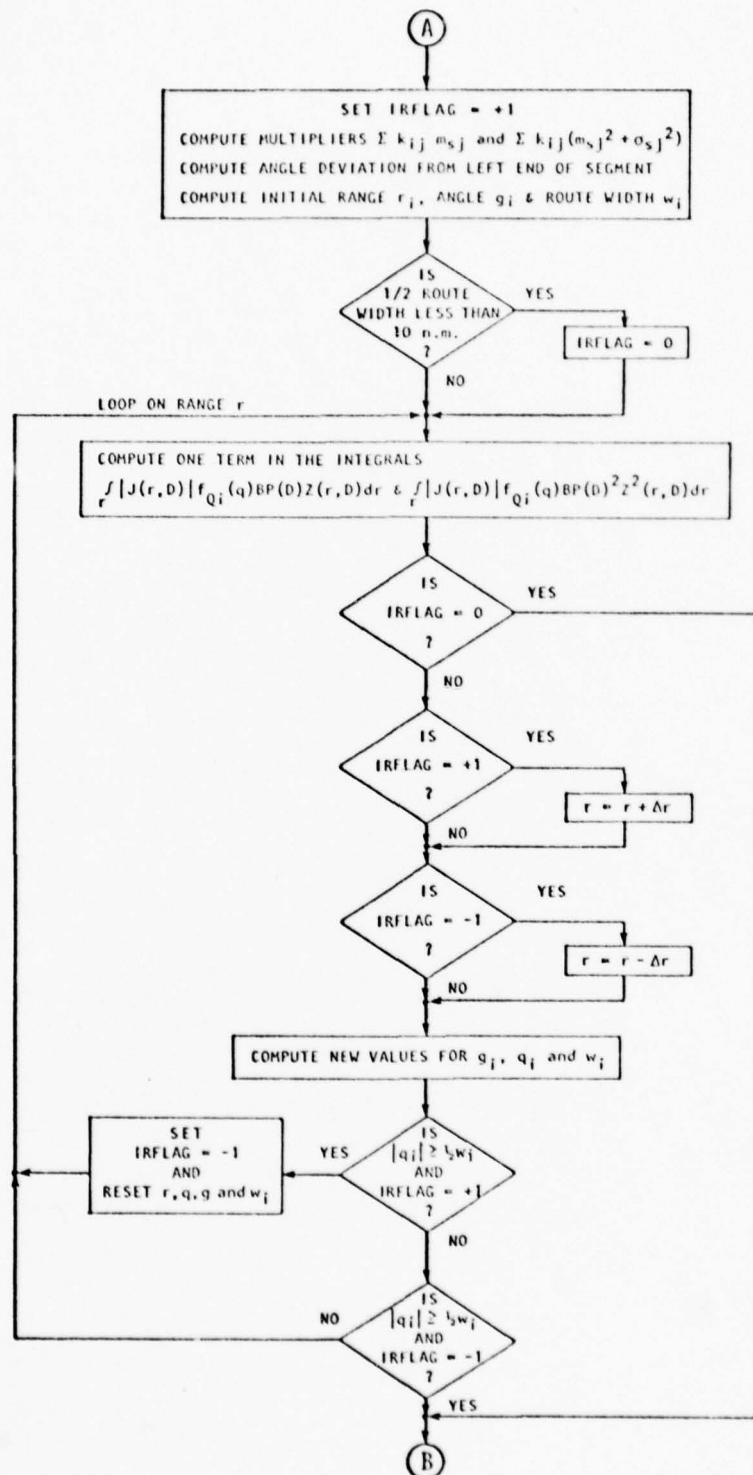
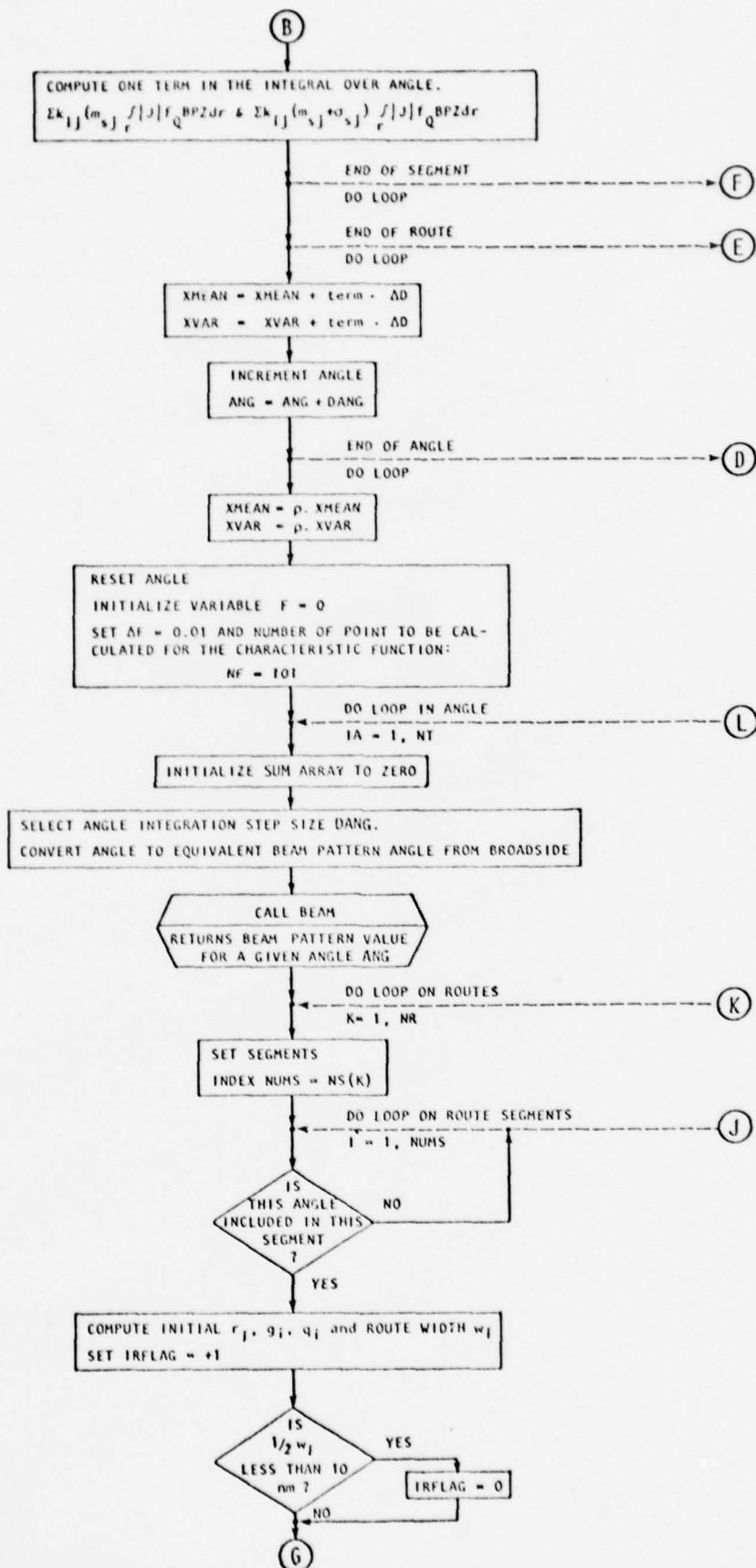
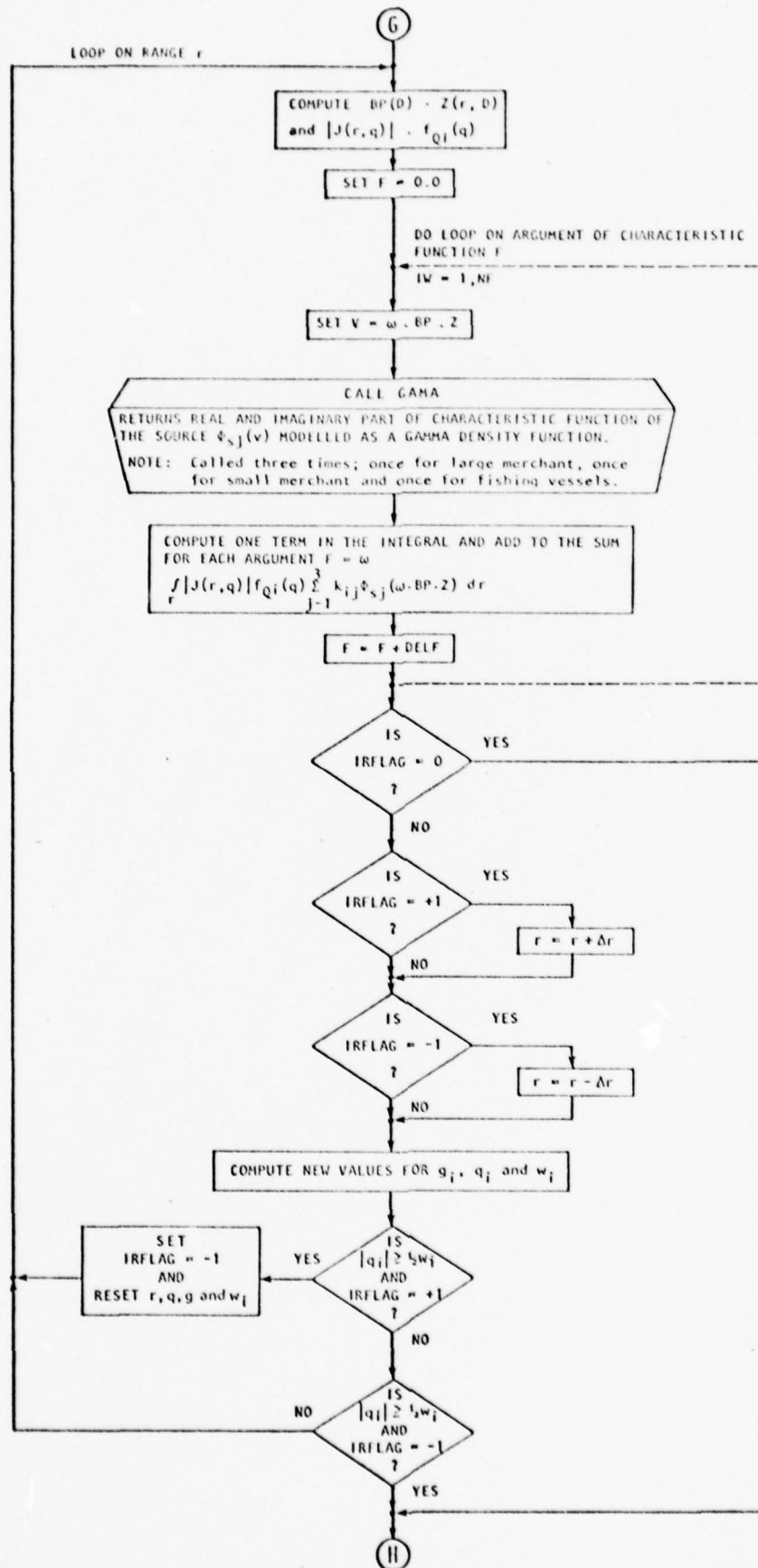
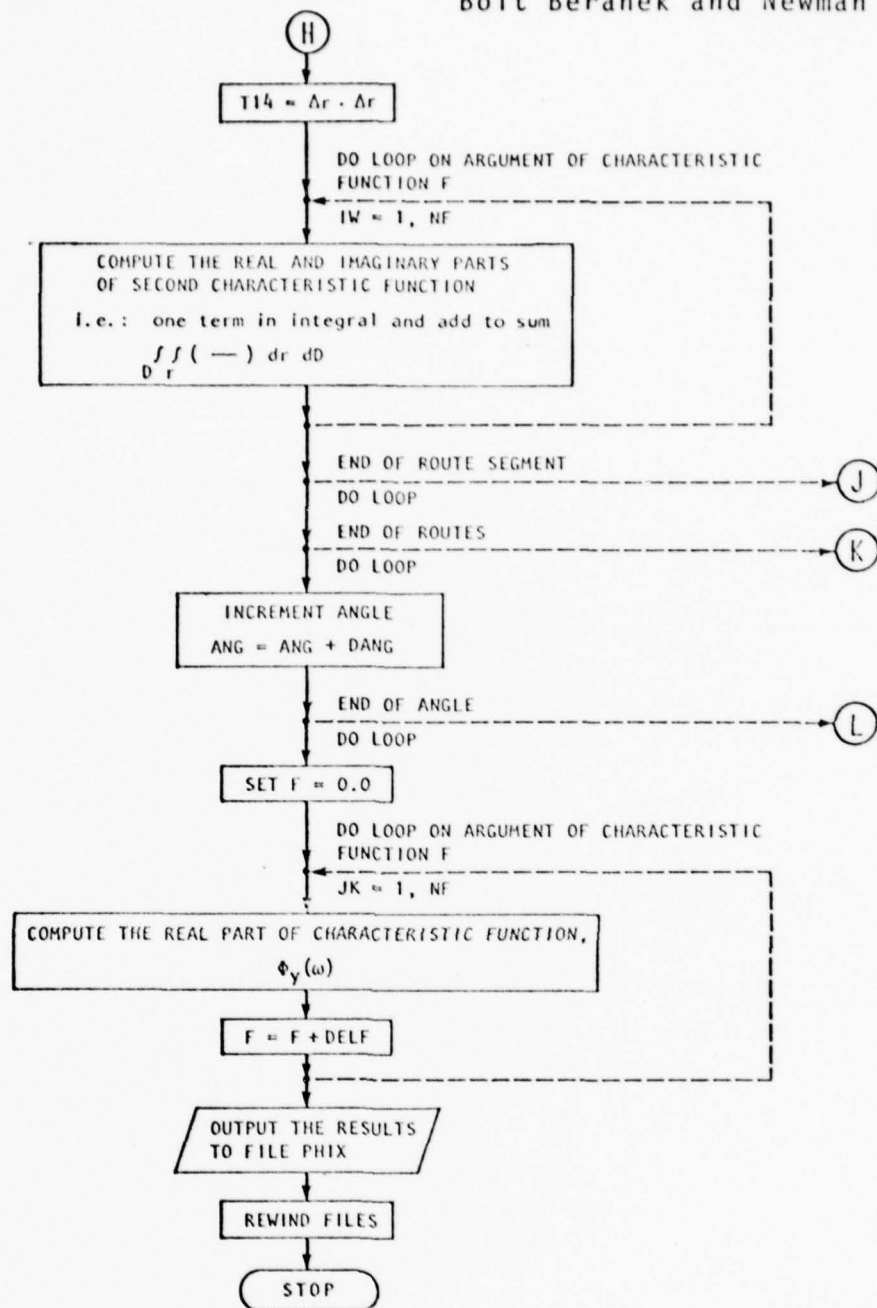


FIGURE 7 Flowchart of Program CFUNC.









The program next enters a section of code which performs the initial calculations. It calculates the two parameters for a gamma probability density function from the means and variances of the sources that were input from file SOURCE for large merchant ships, small merchant ships and fishing vessels. That is, each of the three sources is modeled as a random variable that is gamma distributed. The two parameters of the gamma probability density function are computed as:

$$C = \frac{M_j}{\sigma_j^2}$$

and

$$b+1 = \frac{M_j^2}{\sigma_j^2}$$

where M_j is the mean of the j^{th} source and σ_j^2 is its variance. The program then converts the angles associated with the beam pattern of the array from degrees to radians.

In the final part of the initial calculations, the program computes the parameters needed later in the program for the DO loop on azimuthal angle from the angles associated with the sensors beam pattern. From the steering direction, beamwidth and array broadside angles, the program finds the backbeam. It also tests for an end fire condition. If an end fire condition exists, it divides the angles into two regions--main beam and side lobe. If an end fire condition does not exist, it divides the angles into four regions--main beam, back beam, and two side lobe regions. The program sets the angle integration step size equal to 1/16 the beamwidth in the main lobe and back lobe regions. In the side lobe regions the angle integration step size is set equal to approximately 10 degrees. The program computes the total number of angle integrations NT to cover the complete 360 degrees, which is used in the DO loop on angle. This is the end of the initial section of the program.

The program next enters a section of code which computes the probability of no noise, with variable name DELZ. This term was derived in Section 2.1 of BBN Report 3653, and is computed as

$$\text{DELZ} = \text{EXP} \left\{ - \sum_{i=1}^m \sum_{j=1}^n k_{ij} L_i \right\}$$

where

L_i is the length of the i^{th} route in nautical miles and k_{ij} is the average number of ships of type j per nautical mile of route i .

In the next section of code, the mean and variance of the averaged noise power at the output of a beamformer is computed. The equations were derived in Section 3.1 of BBN Report 3653.

The equations are:

$$m_X = \rho \sum_{i=1}^m \sum_{j=1}^n k_{ij} m_{S_j} \int_{R_i} dD \int dr |J(r,D)| f_{Q_i}(q) z(r,D) \cdot \text{BP}(D)$$

$$\sigma_X^2 = \rho \sum_{i=1}^m \sum_{j=1}^n k_{ij} (m_{S_j}^2 + \sigma_{S_j}^2) \int_{R_i} dD \int dr |J(r,D)| f_{Q_i}(q) z^2(r,D) \cdot \text{BP}^2(D)$$

where m_{S_j} is the average of the mean-squared pressure of sources of type j
 $\sigma_{S_j}^2$ is the variance of the mean-squared pressure of sources of type j .

These are program inputs from file SOURCE.

The above integrals over range r and route segment angle D were approximated by rectangular integration. The integration step size on the range is 10 nautical miles. The integration step size on the angle depends on the region of the beam pattern and was calculated in the initial section of the program. The Jacobian $|J(r,D)|$ is computed from the statement function XJ. The across-route probability density function f_{Q_1} was modeled as a beta probability density function. Its Q_1 value is calculated in Subroutine BETA. The value of transmission loss $Z(r,D)$ is returned to the main program by FUNCTION Z which looks up the value in the transmission loss table TL. The value of the normalized beam pattern is computed in Subroutine BEAM. As seen in Figure 7, the double integrals for the mean and variance are implemented as three nested DO loops. The outer DO loop is on the angle. The DO loop inside this is on the routes. The innermost DO loop is on the route segments. Embedded in the DO loop on the route segments is a loop on the range. Thus, the program evaluates the double integrals for the mean and variance by fixing the angle and evaluating the integral over range across the route segment width to compute one term in the angle integral. It then goes on to the next angle, etc., until it has completed the calculation.

After calculating the mean and variance of the average noise power, the program re-initializes certain parameters that control the numerical integration and enters the section of code that computes the real part of the characteristic function of the average noise power $\phi_Y(\omega)$. The real part of $\phi_Y(\omega)$ is used in Program DENS, described in Section 4 below, to compute the probability density function of average noise power.

The argument of ϕ_Y labeled F in the program, is initialized to zero. The number of points at which the real part of ϕ_Y is computed is set to $NF = 101$, with a step size of $DELF = 0.01$. If more points or a different spacing for the argument of ϕ_Y is desired, the value of NF and $DELF$ must be changed in the program. The real part of ϕ_Y is computed in two basic steps. In the first step, the real and imaginary parts of the second characteristic function $\Psi(\omega)$ are computed from the equation (See Section 2 of BBN Report 3653)

$$\Psi(\omega) = \rho \sum_{i=1}^m \iint_{D \ r_1} |J| f_{Q_1}(q) \sum_{j=1}^3 k_{ij} \phi_{s_j}(\omega \text{ BP } Z_i) \text{ dr } dD$$

where ϕ_{s_j} is the gamma probability density function model for the source of type j .

This computation is accomplished by using rectangular integration to numerically evaluate the double integral. This is done in a set of five nested loops. The outermost loop being on the angle D , while the innermost is on the argument of the characteristic function F .

Once the second characteristic function has been computed, the program computes the real part of the characteristic function from:

$$\text{Real } \phi_Y(\omega) = \text{DELZ} \cdot \text{EXP}(\text{Real } \Psi(w)) \cdot \text{COS}(\text{Imaginary } \Psi(w))$$

for the 101 values of the argument. This concludes the computations sections of Program CFUNK. The resulting values of the real part of $\phi_Y(\omega)$ and the value of w are stored in array PHI.

In the terminal section of the program, the results of Program CFUNK are written to disk file PHIX, which is described in the next section. The input and output files are then rewound, and the programs terminates execution.

3.3 Output File

The output from Program CFUNK to be used in further computations is stored on disk file PHIX. File PHIX is the input file for Program DENS described in Section 4 below. This file contains the header "RESULTS FROM PROGRAM CFUNK" as the first logical record. The remainder of the file contains the values of the parameters for the array; the total number of ships; the probability of no noise; the mean and variance of the beam noise power; the number of sample points that have been computed for the real part of ϕ_Y ; and array PHI containing the real part of $\phi_Y(\omega)$ and w .

Table 5 gives the structure of file PHIX, including the format with which each logical record was written.

Table 5 Structure of File PHIX

(Note: Each line represents one logical record.)

<u>Description</u>	<u>Format</u>
RESULTS from Program CFUNK (header)	(1X, 26H)
Identification Number (IDNUM), frequency (FREQ), frequency band (DELFQ)	(1X, I3, 2F10.4)
Beam Steering angle (BSTER), Beam width (BWIDE)	(1X, 2F15.4)
No. of ships (TSHIP), Probability of "No Noise" (DELZ), XMEAN, XVAR	(1X, E11.4, 3E20.14)
Number of samples (NF), DELF	(1X, I4, E20.14)
Real Part of $\phi_Y(\omega)$ $\left[\text{PHI}(1,1) \right], w \left[\text{PHI}(1,2) \right]$	
PHI(1,2) , PHI(2,2)	
•	•
•	•
•	•
PHI(1,NF) , PHI(2,NF)	
	NF Logical records each containing one row of array PHI

3.4 Glossary and Program Listing

The following glossary contains the definitions of the FORTRAN variable names used in Program CFUNK. The names are presented in alphabetical order.

- | | |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ALPHA | - beam pattern constant which depends on frequency. Program input. |
| ANG | - azimuthal angle from north in radians; initialized to EDGE 1 and used as angle of integration. |
| BACK | - azimuth of back beam. |
| BAZ | - azimuth of array broadside; program input. |
| BP | - value of normalized beam pattern returned from Subroutine BEAM. |
| BSTER | - beam steering angle from north, positive clockwise; program input. |
| BWIDE | - beam width of main lobe of the array; program input. |
| BlF | - (b + 1) parameter for gamma probability density function model of fishing vessel radiated noise, equal to mean squared divided by variance. Used as input to Subroutine GAMMA. |
| BlL | - (b + 1) parameter for gamma probability density function model of large merchant ship radiated noise. |
| BlS | - (b + 1) parameter for gamma probability density function model of small merchant ship radiated noise. |
| CF | - c parameter for gamma probability density function model of fishing vessel radiated noise, equal to mean divided by the variance. |
| CIF | - imaginary part of the characteristic function for fishing vessel radiated noise returned by Subroutine GAMA. |

- CIL - imaginary part of the characteristic function for large merchant ship radiated noise returned by Subroutine GAMA.
- CIS - imaginary part of the characteristic function for small merchant ship radiated noise returned by Subroutine GAMA.
- CL - c parameter for gamma probability density function of large merchant ship radiated noise.
- CS - c parameter for gamma probability density function of small merchant ship radiated noise.
- CRF - real part of the characteristic function for fishing vessel radiated noise returned by Subroutine GAMA.
- CRL - real part of the characteristic function for large merchant ship radiated noise returned by Subroutine GAMA.
- CRS - real part of the characteristic function for small merchant ship radiated noise returned by Subroutine GAMA.
- D - deviation angle from left-end of a segment, in radians; used in numerical integration.
- DANG - step size in deviation angle (ΔD) to be used in numerical integration.
- DANGE - deviation angle step size to be used in end-fire case.
- DANG 1 - deviation angle step size in the main lobe of the beam pattern ; set to 1/16 of the beam width.
- DANG 2 - deviation angle step size to be used in region between main lobe and back lobe going clockwise.
- DANG 3 - deviation angle step size to be used in the back beam; set to 1/16 of beam width.

- DANG 4 - deviation angle step size to be used in region between back lobe and main lobe, going clockwise.
- DELF - step size of the argument of characteristic function ($\Delta\omega$) of beam noise power; fixed in program to 0.01.
- DELFQ - frequency band of received noise in Hertz.
- DELR - integration step size for range r set in radians equivalent to 10 n. miles on earth's surface.
- DELZ - computed probability of no noise.
- D10 - a program constant in radians equal to 10 degrees.
- EDGE 1 - azimuth of left-hand edge of main lobe of beam pattern in radians.
- EDGE 2 - azimuth of right-hand edge of main lobe of beam pattern in radians.
- EDGE 3 - azimuth of left-hand edge of back lobe of beam pattern in radians.
- EDGE 4 - azimuth of right-hand edge of back lobe of beam pattern in radians.
- F - argument ω of the characteristic function of beam noise power $\phi_Y(\omega)$.
- FREQ - center frequency of band of received noise in Hertz.
- G(I,J,K) - an array dimensioned (5x13x10) containing the computed geometric parameters and ship densities for each segment of each route. The index K indicates the route while the index I indicates the segment. For a given segment on a given route the elements in the i th row are:
- G(I,1,K) - route segment left end-point width (radians) W_1 .

- $G(I,2,K)$ - b_i a parameter used to model the route segment width envelope.
- $G(I,3,K)$ - e_i a parameter used in the model of the route segment width envelope.
- $G(I,4,K)$ - $\sin s_i$, sine of the earth centered angle s_i between the sensor and a segment left end-point.
- $G(I,5,K)$ - $\cos s_i$.
- $G(I,6,K)$ - $\tan 1/2 s_i$.
- $G(I,7,K)$ - F_i an interior angle of the triangle formed by the segment and the sensor (in radians).
- $G(I,8,K)$ - Z_i the azimuth of the left end-point of the segment (in radians).
- $G(I,9,K)$ - I_i an interior angle of the triangle formed by the segment and the sensor (in radians).
- $G(I,10,K)$ - length of the segment in nautical miles L_i .
- $G(I,11,K)$ - density of large merchant ships on this segment (ships/n. mile).
- $G(I,12,K)$ - density of small merchant ships on this segment (ships/n. mile).
- $G(I,13,K)$ - density of fishing vessels on this segment (ships/n. mile).
- GI - along-route position variable g_i measured from left end of a segment.
- GIO - initial value of GI.
- IDNUM - an identification number for the particular sensor/route geometry.
- IENDF - flag for end-fire condition of beampattern where
- $IENDF = 0$ Not endfire condition
 $IENDF = 1$ Endfire condition.

- IRFLAG - flag used to determine how and when to reset range loop in range integration where
- IRFLAG = 0 then 1/2 width of the route is less than 10 n.m.
IRFLAG = +1 in upper half of width of route segment
IRFLAG = -1 in lower half of width of route segment.
- ITEST - flag to determine if angle ANG lies in a route segment, returned from function INCLU, where
- ITEST = 0 Not in segment
ITEST = +1 In segment.
- IW - index of DO loop on argument of characteristic function of beam noise power.
- NF - number of sample points at which characteristic function of beam noise power is computed; set to 101 in program.
- NR - number of routes (up to ten).
- NREND - number of angle iterations for numerical integration in endfire case.
- NR1 - number of angle iterations in main lobe.
- NR2 - number of angle iterations between main lobe and back lobe.
- NR3 - number of angle iterations in back lobe.
- NR4 - number of angle iterations between back lobe and main lobe.
- NS - an array dimensioned 10, each element of this array contains the number segments on a route.
- NT - total number of angle iterations in the angle DO loop.
- NUMS - temporary storage for number of segments of a route.

PHI(2,101) - an array dimensioned (2 x 101) which contains the computed values of the real part of the characteristic function $\Phi_Y(\omega)$ and w where:

$$\begin{aligned}\text{PHI}(1,K) &= \text{real part of } \Phi_Y(w) \\ \text{PHI}(2,K) &= w\end{aligned}$$

This array is output to file PHIX.

- PI - π
- Q - across route position variable q_1 ; initially set to zero.
- R - range from sensor; earth centered angle in radians.
- RADS - statement function to convert angles in degrees, minutes and seconds to radians.
- RHO - radius of the earth in nautical miles.
- RO - range from sensor to the center of a route segment (earth centered angle in radians).
- S(3,2) - an array dimensioned (3 x 2) containing the values of the mean and variance of total radiated noise for the three sources, where:
- $$\begin{aligned}S(1,1) &= \text{mean of large merchant ships.} \\ S(1,2) &= \text{variance of large merchant ships.} \\ S(2,1) &= \text{mean of small merchant ships.} \\ S(2,2) &= \text{variance of small merchant ships.} \\ S(3,1) &= \text{mean of fishing vessels} \\ S(3,2) &= \text{variance of fishing vessels.}\end{aligned}$$
- SUM(2,101) - an array dimensioned (2 x 101) used to contain the running sum in the numerical integration over range part of the calculations of the second characteristic function where
- $$\begin{aligned}\text{SUM}(1,K) &= \text{real part} \\ \text{SUM}(2,K) &= \text{imaginary part.}\end{aligned}$$
- TL(351,5) - an array dimensioned (351 x 5) containing the transmission loss table read into the program.
- TPI - 2π .
- TSHIP - Total number of ships

TEMP 1	}	- Temporary storage locations
TEMP 2		
TEMP 3		
T1		
T2		
T3		
T10		
T11		
V	-	argument for source characteristic function used in calling Subroutine GAMA; equal to $w \cdot BP \cdot Z_i$.
WI	-	one half the width of a route segment.
WIDTH	-	statement function to compute an approximation to the route width between route segment end points.
X	-	angle ANG converted to beam pattern coordinates used for input to Subroutine BEAM.
XJ	-	statement function to compute the absolute value of the Jacobian of the transformation, where $J = \sin r_i / \cos q_i.$
XMEAN	-	computed value of the mean of received beam noise power.
XS	-	beam steering angle with respect to array broadside, used as input to Subroutine BEAM.
XVAR	-	computed value of the variance of received beam noise power.

The following is a listing of the FORTRAN code for the main program.

```

      PROGRAM CFUNK(OUTPUT,GEO,TL10H,SOURCE,PHIX,TAPE1=GEO,
+ TAPE2=TL10H,TAPE3=SOURCE,TAPE6=PHIX)
C
C
C PROGRAM TO COMPUTE THE CHARACTERISTIC FUNCTION OF RECEIVED
C NOISE AT THE SENSOR.  WRITTEN JUNE 1978 BY R. ZESKIND.
C
C ----- INPUT FILES -----
C   GEO= CONTAINS ROUTE/SENSOR GEOMETRIC PARAMETERS AND
C   SHIPPING DENSITIES COMPUTED IN GEOMETRY PROGRAM
C   TL10H= CONTAINS TRANSMISSION LOSS DATA
C   SOURCE= CONTAINS SOURCE RADIATED NOISE DATA ,BEAM
C   PATTERN PARAMETERS AND OTHER FREQUENCY DEPENDENT
C   PARAMETERS.
C -----OUTPUT FILE -----
C   PHIX= CONTAINS REAL PART OF CHARACTERISTIC FUNCTION AS
C   COMPUTED BY THIS PROGRAM AND THE MEAN AND VARIANCE.
C   USED AS INPUT TO THE PROBABILITY DENSITY PROGRAM.
C
C
C   DIMENSION G(5,13,10),NS(10),PHI(2,101),S(3,2),SUM(2,101)
C   COMMON/XLOSS/TL(351,5)
C   DATA G/650*0.0/,NS/10*0/,PHI/202*0.0/
C
C STATEMENT FUNCTIONS AND PROGRAM CONSTANTS
C   RADS(D,XM,S)=1.74532952E-02*(D+(XM/60.)+(S/3600.))
C   WIDTH(XG,B,E,XW)=XW+(B*XG)+(E*XG*XG)
C   XJ(R1,Q1)=ABS(SIN(R1)/COS(Q1))
C   DELR=2.9088821E-03
C   RHO=3437.746771
C   PI=3.141592654
C   TPI=2.*PI
C   D10=0.1745329252
C
C
C
C REWIND FILES
C   REWIND1
C   REWIND2
C   REWIND3
C   REWIND6
C
C
C ..

```

```

C  READ IN DATA FROM FILE GEO
      READ(1,10) IDNUM,NR
10    FORMAT(1X,13,2X,13)
      READ(1,15) NS
15    FORMAT(1X,10I3)
      DO 20 K=1,NR
        NUMS=NS(K)
        DO 25 I=1,NUMS
          READ(1,30)(G(I,J,K),J=1,13)
30    FORMAT(1X,4E20.14)
25    CONTINUE
20    CONTINUE
C
C  READ IN TRANSMISSION LOSS DATA
      READ(2,40) T1,T2,T3
40    FORMAT(1X,A10,A10,A5)
      READ(2,50)((TL(K1,K2),K2=1,5),K1=1,351)
50    FORMAT(1X,5E15.8)
C
C  CONVERT FIRST ROW OF TL FROM DEGREES TO RADIANS
      DO 60 J=1,5
        TL(1,J)=RADS(TL(1,J),0.,0.)
60    CONTINUE
C
C  READ IN SOURCE DATA FILE
      DO 70 K=1,3
        READ(3,80) S(K,1),S(K,2)
80    FORMAT(1X,2E20.13)
70    CONTINUE
      READ(3,80) FREQ,DELFQ
      READ(3,85) BAZ,BSTER,ALPHA,BWIDE
85    FORMAT(4E18.10)
C
C  COMPUTE C AND B+1 GAMA DENSITY FUNCTION PARAMETERS
C  FROM SOURCE MEANS AND VARIANCES FOR LARGE, SMALL AND
C  FISHING VESSELS
      CL=S(1,1)/S(1,2)
      B1L=CL*S(1,1)
      CS=S(2,1)/S(2,2)
      B1S=CS*S(2,1)
      CF=S(3,1)/S(3,2)
      B1F=CF*S(3,1)
C
C  CONVERT BAZ ,BSTER AND BWIDE TO RADIANS
      BAZ=RADS(BAZ,0.,0.)
      BSTER=RADS(BSTER,0.,0.)
      BWIDE=RADS(BWIDE,0.,0.)
C
C ***** END OF INPUT SECTION *****
..

```

```
C  INITIAL CALCULATIONS
C    COMPUTE ANGLE INFORMATION
C  COMPUTE AZIMUTH OF MAIN LOBE EDGES E1 AND E2
C
    EDGE1=BSTER-(0.5*BWIDE)
    IF(EDGE1.LT.0.)EDGE1=TPI+EDGE1
    EDGE2=BSTER+0.5*BWIDE
    IF(EDGE2.GE.TPI)EDGE2=EDGE2-TPI
C  FIND THE BEAM STEERING ANGLE WITH RESPECT TO BROADSIDE
C  AND THE STEERING ANGLE FOR SUBROUTINE BEAM:
    TEMP1=BSTER-BAZ
    IF(BAZ.GT.BSTER)TEMP1=TEMP1+TPI
    XS=TEMP1
    IF(TEMP1.GT.PI)XS=XS-TPI
C  TEST FOR ENDFIRE CONDITION
    IENDF=0
    IF(XS.EQ.PI.OR.XS.EQ.-PI)IENDF=1
C  COMPUTE BACK BEAM STEERING ANGLE AND EDGES E3 AND E4
    BACK=PI-TEMP1
    IF(TEMP1.GT.PI)BACK=BACK+TPI
    EDGE3=BACK-0.5*BWIDE
    IF(EDGE3.LT.0)EDGE3=EDGE3+TPI
    EDGE4=BACK+0.5*BWIDE
    IF(EDGE4.GE.TPI)EDGE4=EDGE4-TPI
C  INITIALIZE ANGLE TO EDGE 1 AND SET ANGLE REGION 1 PARAMETERS
    ANG=EDGE1
    DANG1=BWIDE/16.0
    NR1=16
C  COMPUTE PARAMETERS FOR THE REMAINING REGIONS
    IF(IENDF.EQ.1)GOTO 115
C  REGION TWO
    TEMP2=EDGE3-EDGE2
    IF(TEMP2.LT.0.0)TEMP2=TEMP2+TPI
    NR2=INT(TEMP2/D10)
    DANG2=TEMP2/FLOAT(NR2)
C  REGION THREE
    NR3=NR1
    DANG3=DANG1
C  REGION FOUR
    TEMP3=EDGE1-EDGE4
    IF(TEMP3.LT.0.0)TEMP3=TEMP3+TPI
    NR4=INT(TEMP3/D10)
    DANG4=TEMP3/FLOAT(NR4)
C  COMPUTE PARAMETERS FOR ANG DO LOOP
    NT=NR1+NR2+NR3+NR4
    GOTO 118
C  ENDFIRE CASE
115  TEMP3=TPI-BWIDE
    NREND=INT(TEMP3/D10)
    DANGE=TEMP3/FLOAT(NREND)
    NT=NR1+NREND
118  CONTINUE
C*****END OF INITIAL SECTION*****
..
```

```

C  COMPUTE THE PROBABILITY OF NO NOISE, TERM DELZ
    TSHIP=0.0
    DO 120 K=1, NR
        NUMS=NS(K)
        DO 125 I=1, NUMS
            TSHIP=TSHIP+G(I, 10, K)*(G(I, 11, K)+G(I, 12, K)+G(I, 13, K))
125    CONTINUE
120    CONTINUE
    DELZ=EXP(-TSHIP)
C
C
C  SECTION TO COMPUTE MEAN AND VARIANCE
    XMEAN=0.0
    XVAR=0.0
    DO 200 IA=1, NT
        T10=0.0
        T11=0.0
C  SELECT WHICH DANG TO USE DEPENDING ON WHICH REGION
C  YOU ARE IN AND WHICH CASE
        IF(LENDF.EQ.1) GOTO 230
        DANG=DANG1
        IF(IA.GT.NR1) DANG=DANG2
        IF(IA.GT.(NR1+NR2)) DANG=DANG3
        IF(IA.GT.(NR1+NR2+NR3)) DANG=DANG4
        GOTO 240
230    DANG=DANG1
        IF(IA.GT.NR1) DANG=DANG2
240    CONTINUE
C  CONVERT ANG INTO EQUIVALENT BEAM PATTERN ANGLE
C  FROM BROADSIDE
        X=ANG-BAZ
        IF(BAZ.GT.ANG) X=X+TPI
        IF(X.GT.PI) X=X-TPI
        CALL BEAM(BP, ALPHA, X, XS)
        DO 210 K=1, NR
            NUMS=NS(K)
            DO 220 I=1, NUMS
C  TEST FOR INCLUSION OF SEGMENT FOR THIS ANGLE
                TEMP1=G(I, 8, K)+G(I, 9, K)
                IF(TEMP1.GE.TPI) TEMP1=TEMP1-TPI
                ITEST=INCLU(G(I, 8, K), TEMP1, ANG)
                IF(ITEST.EQ.0) GOTO 220
C  COMPUTE MULTIPLIERS
                TEMP2=G(I, 11, K)*S(1, 1)+G(I, 12, K)*S(2, 1)+G(I, 13, K)*S(3, 1)
                TEMP3=G(I, 11, K)*(S(1, 1)*S(1, 1)+S(1, 2))
                1  +G(I, 12, K)*(S(2, 1)*S(2, 1)+S(2, 2))
                2  +G(I, 13, K)*(S(3, 1)*S(3, 1)+S(3, 2))
                ..

```



```

C  COMPUTE ANGLE DEVIATION D FROM LEFT END OF SEGMENT
    D=ANG-G(I,8,K)
    IF(D.LT.0.0)D=D+TPI
    T13=SIN(D)
    T12=COS(D)
C  COMPUTE INITIAL RANGE ANGLE RO FOR THE GIVEN
C  ANGLE D AND INITIAL GIO
    IRFLAG=1
    Q=0.0
    TEMP5=(G(I,7,K)-D)*0.5
    TEMP6=(G(I,7,K)+D)*0.5
    TEMP7=G(I,6,K)*SIN(TEMP5)/SIN(TEMP6)
    TEMP8=G(I,6,K)*COS(TEMP5)/COS(TEMP6)
    RO=ATAN(TEMP7)+ATAN(TEMP8)
    GIO=ATAN(TEMP8)-ATAN(TEMP7)
    WI=0.5*WIDTH(GIO,G(I,2,K),G(I,3,K),G(I,1,K))
    IF(WI.LT.DELR) IRFLAG=0
    R=RO
    TEMP5=0.0
C  LOOP ON RANGE R FOR THIS ANGLE AND SEGMENT
    TEMP6=0.0
    GI=GIO
300  TEMP7=Z(R,ANG)*BP
    TEMP8=XJ(R,Q)*BETA(Q,WI)
    TEMP5=TEMP5+TEMP8*TEMP7
    TEMP6=TEMP6+TEMP8*TEMP7*TEMP7
    IF(IRFLAG.EQ.0) GOTO 350
C  INCREMENT RANGE
    IF(IRFLAG.EQ.1)R=R+DELX
    IF(IRFLAG.EQ.-1)R=R-DELX
C  COMPUTE NEW GI, QI, WI
    TEMP7=G(I,5,K)*COS(R)+G(I,4,K)*SIN(R)*T12
    TEMP8=ACOS(TEMP7)
    TEMP7=ASIN(SIN(R)*T12/SIN(TEMP8))
    GI=ATAN(COS(TEMP7-G(I,7,K))*TAN(TEMP8))
    Q=ASIN(SIN(TEMP7-G(I,7,K))*SIN(TEMP8))
    WI=0.5*WIDTH(GI,G(I,2,K),G(I,3,K),G(I,1,K))
C  TEST TO SEE IF NEW VALUE OF QI PUTS YOU OUT OF
C  THE ROUTE SEGMENT ENVELOPE.
    TEMP8=ABS(Q)-WI
    IF(TEMP8.GE.0.0.AND.IRFLAG.EQ.1) GOTO 370
    IF(TEMP8.GE.0.0.AND.IRFLAG.EQ.-1) GOTO 350
    GOTO 300
370  R=RO
    Q=0.0
    GI=GIO
    IRFLAG=-1
    WI=0.5*WIDTH(GI,G(I,2,K),G(I,3,K),G(I,1,K))
    GOTO 300
350  T10=T10+TEMP5*TEMP2*DELX
    T11=T11+TEMP6*TEMP3*DELX
220  CONTINUE
C  END OF SEGMENT LOOP
210  CONTINUE
C  END OF ROUTE LOOP
..

```

```

      XMEAN=XMEAN+T10*DANG
      XVAR=XVAR+T11*DANG
      ANG=ANG+DANG
      IF(ANG.GE.TPI)ANG=ANG-TPI
200  CONTINUE
C   END OF ANGLE LOOP
      XMEAN=RHO*XMEAN
      XVAR=RHO*XVAR
C*****END OF MEAN AND VARIANCE COMPUTATIONS*****
C*****
C   MAIN COMPUTATION
C   COMPUTE REAL PART OF CHARACTERISTIC FUNCTION
C*****
C   RESET ANGLES AND INITIALIZE F PARAMETERS
      ANG=EDGE1
      F=0.0
      DELF=0.01
      NF=101
C*****
C   LOOP ON ANGLE ANG
      DO 400 IA=1,NT
C   INITIALIZE SUM ARRAY TO ZERO
      DO 401 KK=1,NF
        SUM(1,KK)=0.0
        SUM(2,KK)=0.0
401  CONTINUE
C   SELECT WHICH DANG AND WHICH CASE
      IF(IENDF.EQ.1) GOTO 410
      DANG=DANG1
      IF(IA.GT.NR1) DANG=DANG2
      IF(IA.GT.(NR1+NR2)) DANG=DANG3
      IF(IA.GT.(NR1+NR2+NR3)) DANG=DANG4
      GOTO 420
410  DANG=DANG1
      IF(IA.GT.NR1) DANG=DANG2
420  CONTINUE
C   CONVERT ANG INTO EQUIVALENT BEAM PATTERN ANGLE X
C   FROM BROADSIDE AND COMPUTE BP
      X=ANG-BAZ
      IF(PAZ.GT.ANG)X=X+TPI
      IF(X.GT.PI)X=X-TPI
      CALL BEAM (BP,ALPHA,X,XS)
C   LOOP ON ROUTES
      DO 430 K=1,NR
        NUMS=NS(K)
C   LOOP ON SEGMENTS
        DO 440 I=1,NUMS
C   TEST FOR INCLUSION OF ANG IN THIS SEGMENT
          TEMP1=G(I,8,K)+G(I,9,K)
          IF(TEMP1.GE.TPI)TEMP1=TEMP1-TPI
          ITEST=INCLU(G(I,8,K),TEMP1,ANG)
          IF(ITEST.EQ.0)GOTO 440
C   COMPUTE ANGLE DEVIATION D FROM LEFT END OF SEGMENT
          D=ANG-G(I,8,K)
          IF(D.LT.0.0)D=D+TPI
          T13=SIN(D)
          T12=COS(D)

```

..

```

C  COMPUTE INITIAL RANGE RO AND GIO
    IRFLAG=1
    Q=0.0
    TEMP5=(G(I,7,K)-D)*0.5
    TEMP6=(G(I,7,K)+D)*0.5
    TEMP7=G(I,6,K)*SIN(TEMP5)/SIN(TEMP6)
    TEMP8=G(I,6,K)*COS(TEMP5)/COS(TEMP6)
    RO=ATAN(TEMP7)+ATAN(TEMP8)
    GIO=ATAN(TEMP8)-ATAN(TEMP7)
    WI=0.5*WIDTH(GIO,G(I,2,K),G(I,3,K),G(I,1,K))
    IF(WI.LT.DELR)IRFLAG=0
    R=RO
    GI=GIO
C  LOOP ON RAGE R FOR FIXED ANGLE
450    TEMP7=BP*Z(R,ANG)
        TEMP8=XJ(R,Q)*BETA(Q,WI)
        F=0.0
C  LOOP ON CHARACTERISTIC FUNCTION ARGUMENT F
    DO 460 IW=1,NF
        V=F*TEMP7
        CALL GAMA(V,CL,B1L,CRL,CIL)
        CALL GAMA(V,CS,B1S,CRS,CIS)
        CALL GAMA(V,CF,B1F,CRF,CIF)
        SUM(1,IW)=SUM(1,IW)+TEMP8*(G(I,11,K)*CRL
1      +G(I,12,K)*CRS+G(I,13,K)*CRF)
        SUM(2,IW)=SUM(2,IW)+TEMP8*(G(I,11,K)*CIL
1      +G(I,12,K)*CIS+G(I,13,K)*CIF)
        F=F+DELF
460    CONTINUE
C  INCREMENT RANGE
    IF(IRFLAG.EQ.0) GOTO 470
    IF(IRFLAG.EQ.1) R=R+DELX
    IF(IRFLAG.EQ.-1)R=R-DELX
C  COMPUTE NEXT GI, QI AND WI
    TEMP7=G(I,5,K)*COS(R)+G(I,4,K)*SIN(R)*T12
    TEMP8=ACOS(TEMP7)
    TEMP7=ASIN(SIN(R)*T13/SIN(TEMP8))
    GI=ATAN(COS(TEMP7-G(I,7,K))*TAN(TEMP8))
    Q=ASIN(SIN(TEMP7-G(I,7,K))*SIN(TEMP8))
    WI=0.5*WIDTH(GI,G(I,2,K),G(I,3,K),G(I,1,K))
C  TEST TO SEE IF NEW VALUE OF QI PUTS YOU OUT OF THE ROUTE
C  SEGMENT ENVELOPE
    TEMP8=ABS(Q)-WI
    IF(TEMP8.GE.0.0.AND.IRFLAG.EQ.1) GOTO 480
    IF(TEMP8.GE.0.0.AND.IRFLAG.EQ.-1) GOTO 470
    GOTO 450
480    R=RO
        Q=0.0
        GI=GJO
        IRFLAG=-1
        WI=0.5*WIDTH(GI,G(I,2,K),G(I,3,K),G(I,1,K))
    GOTO 450
470    T14=DELX*DANG
        DO 490 IW=1,NF
            PHI(1,IW)=PHI(1,IW)+SUM(1,IW)*T14
            PHI(2,IW)=PHI(2,IW)+SUM(2,IW)*T14
490    CONTINUE
440    CONTINUE
C*****END OF SEGMENT LOOP*****

```

```

430    CONTINUE
C*****END OF ROUTE LOOP*****
      ANG=ANG+DANG
      IF(ANG.GE.TPI)ANG=ANG-TPI
400    CONTINUE
C*****END OF ANGLE LOOP*****
C  COMPUTE REAL PART OF CHARACTERISTIC FUNCTION FROM
C  SECOND CHARACTERISTIC FUNCTION
      F=0.0
      DO 500 JK=1,NF
        PHI(1,JK)=DELZ*EXP(RHO*PHI(1,JK))*COS(RHO*PHI(2,JK))
        PHI(2,JK)=F
        F=F+DELF
500    CONTINUE
C
C*****END OF CHARACTERISTIC FUNCTION COMPUTATIONS*****
C
C  OUTPUT RESULTS TO FILE PHIX
C
C  WRITE HEADER ON FILE PHIX
      WRITE(6,510)
510    FORMAT(1X,26HRESULTS FROM PROGRAM CFUNK)
      WRITE(6,520) IDNUM,FREQ,DELFQ
520    FORMAT(1X,I3,2F10.4)
C
C  CONVERTS BSTER AND BWIDE TO DEGREES FROM RADIANS
C
      BSTER=180.*BSTER/PI
      BWIDE=180.*BWIDE/PI
      WRITE(6,530)BSTER,BWIDE
530    FORMAT(1X,2F15.4)
C
      WRITE(6,540)TSHIP,DELZ,XMEAN,XVAR
540    FORMAT(1X,E11.4,3E20.14)
      WRITE(6,545)NF,DELF
545    FORMAT(1X,I4,E20.14)
      WRITE(6,550)(PHI(1,K),PHI(2,K),K=1,NF)
550    FORMAT(1X,E20.14,3X,E20.14)
C
C
C  REWIND FILES
      REWIND 1
      REWIND 2
      REWIND 3
      REWIND 6
C
      STOP
      END
..

```

3.5 Subprograms

This section describes the subroutine and function subprograms that are called from the main program of CFUNK.

3.5.1 Function Z

This function subprogram is used to return a value of transmission loss to the main program from input values of range and azimuth. The array TL containing the transmission loss table is stored and labeled common XLOSS. The calling sequence is:

Function Z (R, D)

where Z = returned value of transmission loss in power

R = range as earth centered angle in radians
which is converted to n.m. in this function

D = azimuth in radians.

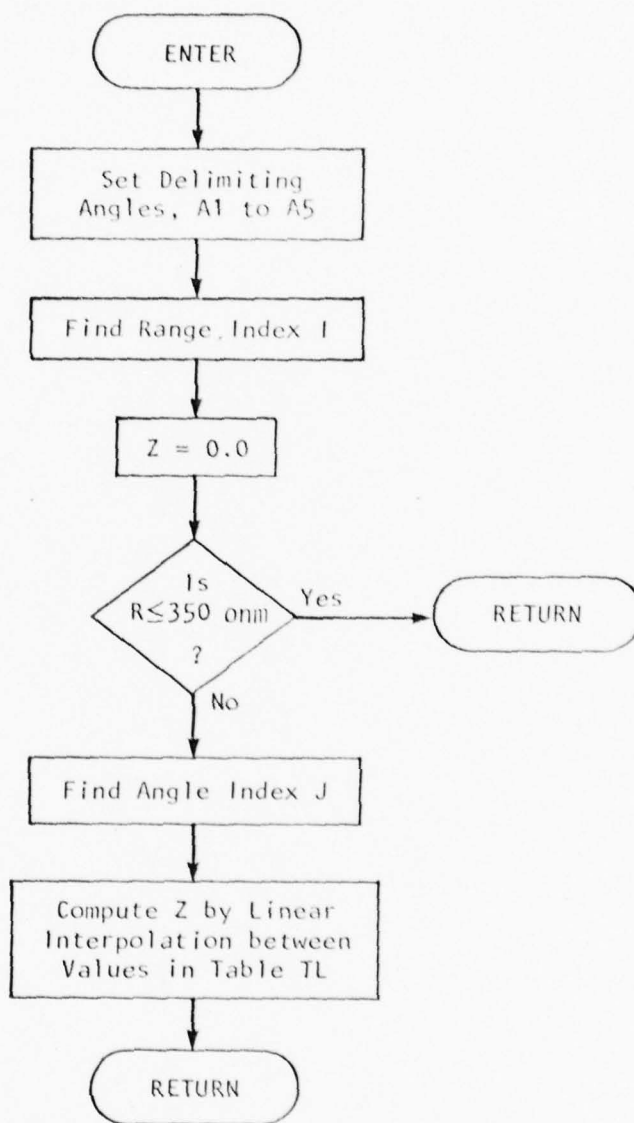
Figure 8 presents a flowchart for Function Z. Upon entry, five delimiting angles A1 to A5 in radians are set to divide up the azimuth into the five regions where each of the columns of array TL are to be used. For example, if the transmission loss is given at the five azimuthal angles 69° , 132° , 285° , 294° and 351° (where 69° is the first column of TL and 351° the last) then choose $A1 = 30^\circ$, $A2 = 100.5^\circ$, $A3 = 208.5^\circ$, $A4 = 289.5^\circ$ and $A5 = 322.5^\circ$ as the delimiting angles. Note that the programmer must set the values of A1 to A5 in the program before execution.

Next, the function determines the row index I from

$$I = (\text{range in n.m.})/10 + 1$$

The function sets $Z = 0$ and tests to see if the index $I > 350$; that is if the range is greater than 3500 miles. If this is the case, the function returns to the main program.

If not, the program next tests to determine the two values of delimiting angles that the angle D ties between, and thus selects the appropriate column index J. Once I and J are found, the function then computes a value of transmission loss Z by linear interpolation on the range. The value of Z is then returned to the main program.

FIGURE 8 Flowchart Function $Z(R, D)$.

The listing of the FORTRAN code for Function Z is as follows:

```
      FUNCTION Z(R,D)
      COMMON/XLOSS/TL(351,5)
C
C RETURNS A TRANSMISSION LOSS FROM THE TABLE TL
C TL IS THE INPUT TRANSMISSION LOSS TABLE FOR FIVE
C AZIMUTH ANGLES. TEN N.M. INCREMENTS IN RANGE
C Z = RETURNED VALUE OF TRANSMISSION LOSS
C R = RANGE
C D = ANGLE FROM NORTH
C
      RADS(V)=V*(.0174532925)
      RHO=3437.746771
C CONVERT RANGE FROM RADIANS TO N.M.
      RX=R*RHO
C SET DELIMITING ANGLES
      A1=RADS(30.)
      A2=RADS(100.5)
      A3=RADS(208.5)
      A4=RADS(289.5)
      A5=RADS(322.5)
C
C FIND RANGE INDEX I (ROW)
      I=INT(RX/10.)+1
      Z=0.
      IF(I.GE.350) RETURN
C
C FIND ANGLE INDEX J (COLUMN)
      IF(D.LT.A2.AND.D.GE.A1) J=1
      IF(D.LT.A3.AND.D.GE.A2) J=2
      IF(D.LT.A4.AND.D.GE.A3) J=3
      IF(D.LT.A5.AND.D.GE.A4) J=4
      IF(D.LT.A1.OR.D.GT.A5) J=5
C
C INTERPOLATE OVER RANGE
C
      SLOPE=(TL(I+1,J)-TL(I,J))/10.
      RI=(FLOAT(I)-1.)*10.
      Z=TL(I,J)+(RX-RI)*SLOPE
      RETURN
      END
```

3.5.2 Function BETA

The computation of the characteristic function $\Phi_Y(\omega)$ requires a probability density function for the ship position across the route. In Section 2 of BBN Report No. 3653 this density function is denoted for the i th route by $f_{Q_i}(q)$, where q is the earth centered angle in radians representing ship position measured from the center of the route.

Define the width of the route, at a given point along the route, as W where W is an earth centered angle in radians. Define the half-width of the route as

$$q_0 = W/R$$

Therefore, the probability of a ship being positioned outside the interval $-q_0 \leq q \leq +q_0$ is zero.

Since exact information of ship distributions across the world's shipping routes is not available at present, a probability density function must be chosen to model the uncertainty in ship position within the route width W . An appropriate choice is felt to be the BETA probability density function with the parameter values both set equal to two. This density function is symmetric about its mean value, which if chosen as the center of the route, does not favor one side of a route over the other. The density function to be used for ships' position across the route is

$$f_{Q_i}(q) = \begin{cases} \frac{1.875}{2q_0} \left(1 - \left(\frac{q}{q_0} \right)^2 \right)^2, & \text{for } -q_0 \leq q \leq +q_0 \\ 0, & \text{otherwise} \end{cases}$$

Figure 9 shows a plot of the density function.

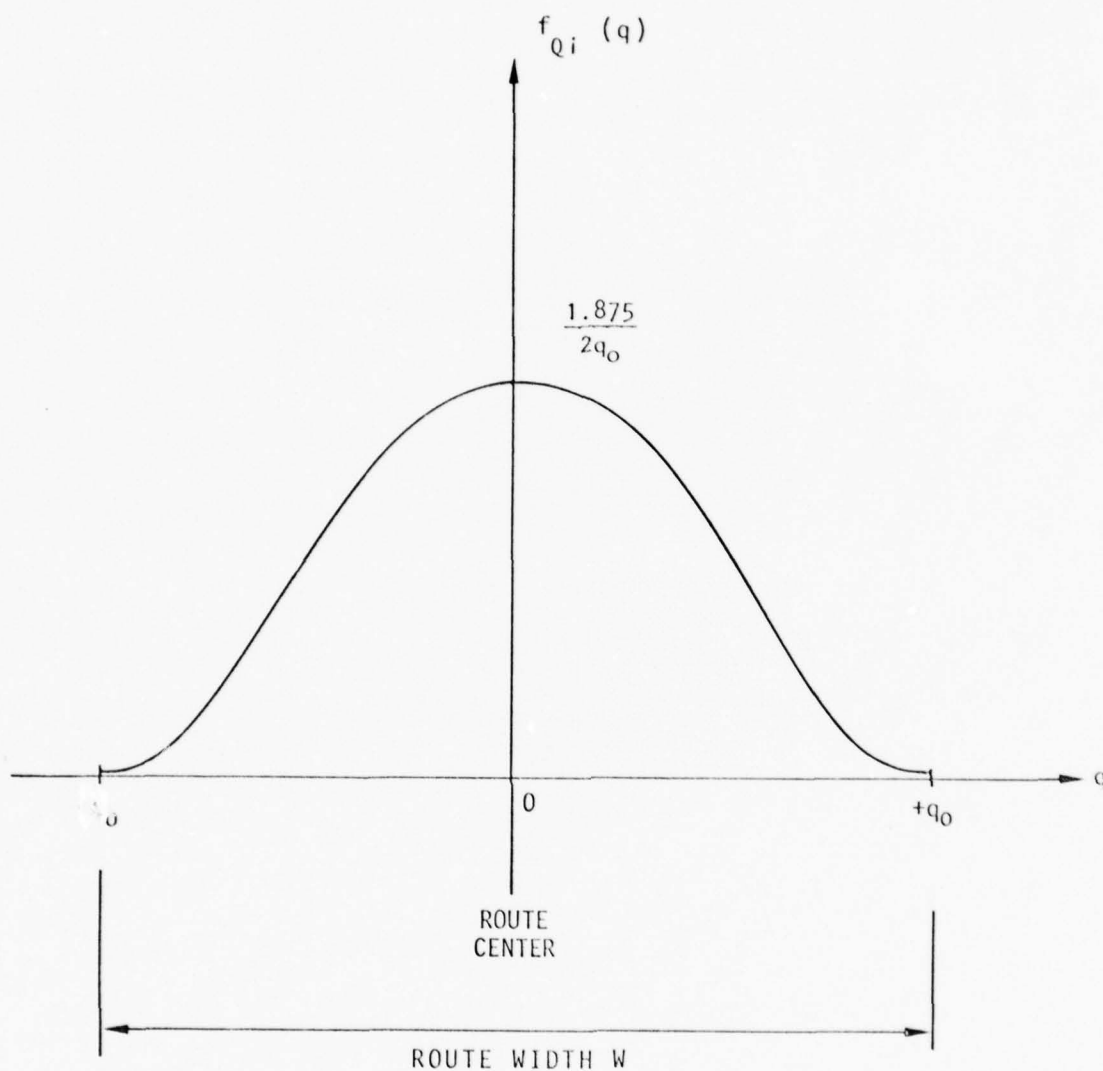


FIGURE 9 Across-Route Density Function.

Function BETA was written in FORTRAN to return a value of $f_{Q_1}(q)$ from inputs q and q_0 . The calling sequence for this function is BETA(Q, Q_0) where

BETA - $f_{Q_1}(q)$; returned value of density function.

Q - q in radians

Q_0 - $1/2$ route width in radians.

Figure 10 presents the flowchart for Function BETA. The function listing is as follows:

```

      FUNCTION BETA(Q,Q0)
      C TRANSVERSE SHIP POSITION DENSITY FUNCTION
      C BETA = RETURNED VALUE OF DENSITY FUNCTION
      C Q = ACROSS-ROUTE EARTH CENTERED ANGLE (RADS.)
      C Q0 = ONE HALF THE ROUTE WIDTH
      C
      BETA=0.0
      IF(Q.LT.-Q0.OR.Q.GT.Q0) RETURN
      BETA=1.0
      IF(Q0.EQ.0.0) RETURN
      X=Q/Q0
      X=1.0-(X*X)
      BETA=(1.875*X*X)/(2.*Q0)
      RETURN
      END

```

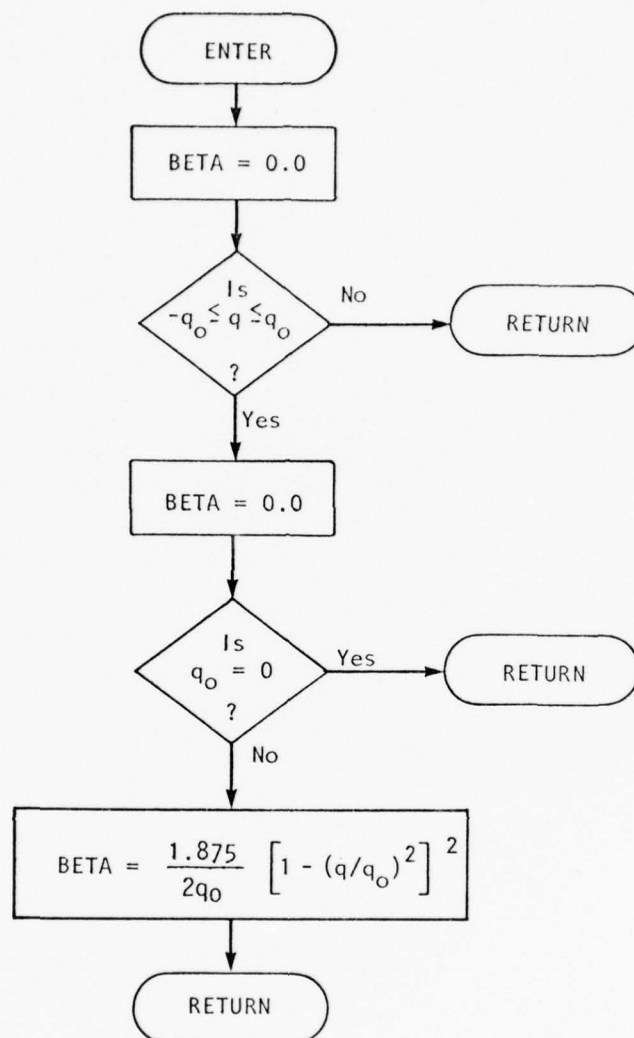



FIGURE 10 Flowchart for Function BETA

3.5.3 Function INCLU

This function is used to determine if an angle C is included between angles A and B. Angles A, B and C are azimuthal angles in radians. It is assumed that A is always counter clockwise from B. The calling sequence is

INCLU (A,B,C)

where

the returned value of INCLU = +1 if C is included between A and B, and zero otherwise.

Figure 11 presents a flowchart for FUNCTION INCLU. The listing of the function is as follows:

```
FUNCTION INCLU(A,B,C)
C FUNCTION DETERMINES IF ANGLE C IS INCLUDED
C BETWEEN ANGLES A AND B.
C INCLU= 0 IF NOT INCLUDED
C INCLU= 1 IF INCLUDED
C
  INCLU=0
  IF(B.GT.A.AND.C.GE.A.AND.B.GE.C) INCLU=1
  IF(A.GT.B.AND.B.GE.C) INCLU=1
  IF(A.GT.B.AND.C.GE.A) INCLU=1
  RETURN
END
```

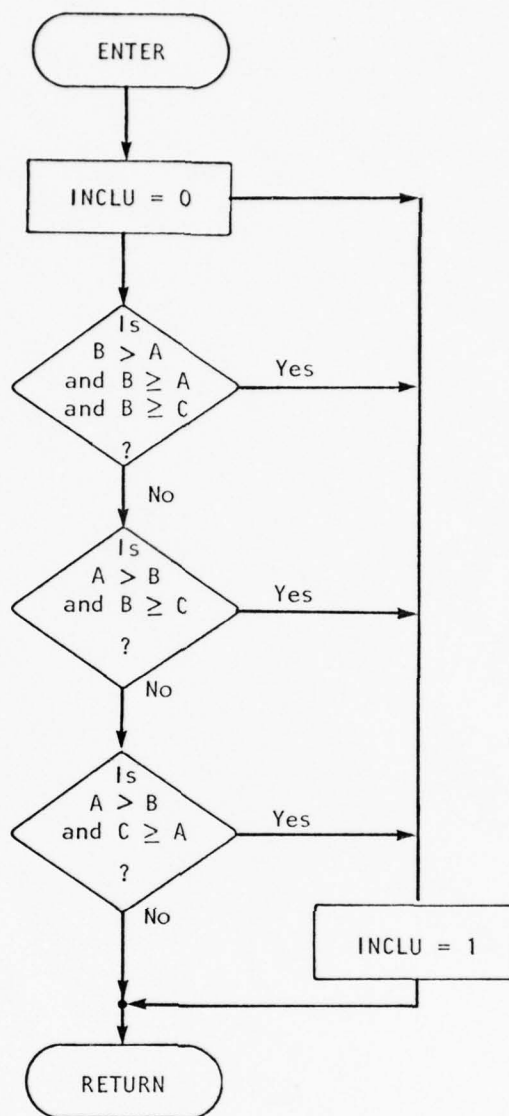


FIGURE 11 Flowchart for Function INCLU.

3.5.4 Subroutine GAMA

This subroutine computes the real and imaginary part of the characteristic function of the gamma probability density function. The gamma probability density function is used to statistically model the radiated noise from merchant ships. The characteristic function is given by

$$\Phi(\omega) = \left[\frac{C}{C - j\omega} \right]^{b+1}$$

where

$$C = M/\sigma^2$$

$$b + 1 = M^2/\sigma^2$$

$$M = \text{Mean}$$

$$\sigma^2 = \text{Variance}$$

$$\text{and } j = \sqrt{-1}$$

The two parameters C and $b + 1$, computed in the main program from the source mean and variance, are inputs to the subroutine. The characteristic function can be written in terms of its real and imaginary parts as

$$\text{Real } \Phi(\omega) = \left[\frac{C}{\sqrt{C^2 + \omega^2}} \right]^{b+1} \cos \left\{ (b+1) \text{Arctan } (\omega/C) \right\}$$

and

$$\text{IM } \Phi(\omega) = \left[\frac{C}{\sqrt{C^2 + \omega^2}} \right]^{b+1} \sin \left\{ (b+1) \text{Arctan } (\omega/C) \right\} ,$$

The calling sequence for the subroutine is GAMA (W, C, D, CR, CI) where the inputs are:

W = argument ω of characteristic function

C = parameter C of the gamma probability density function

D = b + 1 parameter

and the outputs are:

CR = real part of characteristic function

CI = imaginary part of characteristic function.

Figure 12 presents a flowchart for Subroutine GAMA. The listing of the program is as follows:

```
      SUBROUTINE GAMA(W,C,D,CR,CI)
C COMPUTES THE CHARACTERISTIC FUNCTION OF GAMMA DENSITY
C W= FREQ. IN RADS.
C C= PARAMETER C OF DENSITY
C D= B+1 PARAMETER OF DENSITY
C CR= REAL PART
C CI= IMAGINARY PART
      X=W/C
      A=(1.0/SQRT(1.0+X*X))**D
      Y=D*ATAN(X)
      CR=A*COS(Y)
      CI=A*SIN(Y)
      RETURN
      END
```

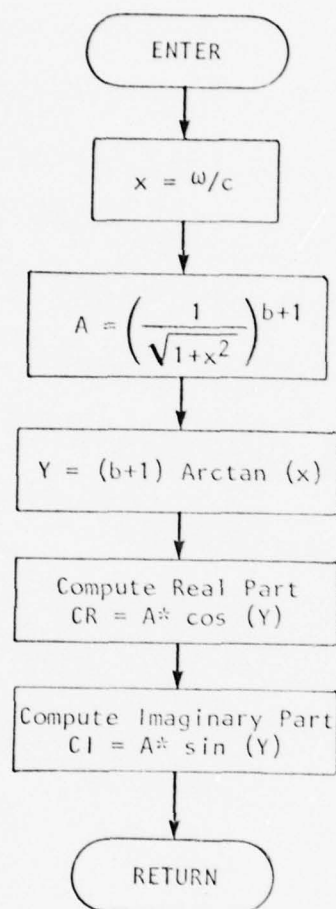



FIGURE 12 Subroutine GAMA Flowchart.

3.5.5 Subroutine Beam

In Section 4.1 of BBN Report 3653 a normalized beam pattern model was described in detail. The normalized beam pattern values is needed in the main program, where it is used to weight the transmission loss values.

This subroutine was written to compute the normalized beam pattern model. The calling sequence is SUBROUTINE BEAM (BP, ALPHA, X, XS) where

BP = value of normalized beam pattern to be returned.

ALPHA = value of frequency dependent constant which is a program input.

X = angle in radians with respect to array broadside in radians, which is an input to the main program.

Figure 13 presents the flowchart for Subroutine BEAM. A listing is as follows:

C

```

SUBROUTINE BEAM(BP,ALPHA,X,XS)
PI=3.14159265
BETA=ALPHA*(SIN(X)-SIN(XS))
IF(BETA.EQ.0.) GO TO 150
AVAL=ABS(BETA)
IF(AVAL.GT.2.) GO TO 100
IF(AVAL.EQ.1.) GO TO 200
TEMP1=SIN(PI*BETA)
TEMP1=TEMP1*TEMP1
TEMP2=1.0/((PI*BETA)*(1.0-BETA*BETA))
BP=TEMP1*TEMP2*TEMP2
GO TO 250
100  BP=1.0/((PI*BETA)*(1.0-BETA*BETA))
    BP=0.5*BP*BP
    GO TO 250
150  BP=1.
    GO TO 250
200  BP=0.25
250  RETURN
END

```

4. PROGRAM DENS

The final computer program in the implementation of the Ambient Noise Model is Program DENS.

The Ambient Noise Model produces as output statistical measures of the noise power at the output of a beamformer. The measures that are of interest are the probability density function of output noise power (in power), the density function of output noise power (in db re/watt), and the distribution function of output noise power (in db).

Program DENS calculates the probability density function for the beamformer output noise power from the real part of its characteristic function. This inverse transformation was derived and discussed in Section 3.1 of BBN Report 3653. One density function is found as a function of power (watts) and another as a function of db (re/watt). Next, the program calculates the distribution function (in db) from the density function in power. The program reads input data from the previously created disk file PHIX and prints the results at the terminal.

4.1 Input Data File

The input data for Program DENS is read from disk file PHIX. The contents and structure of the logical records in file PHIX were described in Section 3.3 above.

4.2 Main Program

The main program of DENS performs the following functions:

- 1) initialize arrays and set program constants
- 2) reads input data from file PHIX
- 3) computes the required probability density and distribution functions
- 4) prints the results at the terminal.

Program DENS calls three subroutines from the main program during its execution. These subroutines are

- IFT - Calculates the inverse fourier transform, GX, for a given value of X from the real part of the characteristic function in array R. Array R, the number of sample points JLM, and the sample point spacing DELWL are stored in the labeled common area COM.

- POW2DB - Converts a probability density function in power into a probability density function in db/re/watt.
- DIST - Calculates the distribution function in power from an input density function in power.

These subroutines are described in Section 4.5 below.

4.2.1 Common Areas

Program DENS makes use of one common area called COM. COM contains the real array R, a real constant DELW1, and an integer constant J1M. COM is used by the utility subroutine IFT. Array R contains the real part of a characteristic function. J1M is the number of sample points in the characteristic function, and DELW1 is the sample point spacing.

4.2.2 Description and Flowchart

Figure 14 presents a flowchart for Program DENS. The program starts by initializing constants and arrays. Next, DENS reads data from the input data file PHIX. These items are the header; an identification number, center frequency, and bandwidth; the beam steering angle and major lobe width; the total mean number of ships, the probability of no noise, the mean noise power, and the variance in noise power; the number of sample points and the sample spacing of the characteristic function; and finally, the real part of the characteristic function stored in array R.

Now, the first of three functions is calculated. The first function is the probability density, $f(X)$, for the noise power at the output of a beamformer (in power). This function starts at $X = 0$. The calculation proceeds in two steps; first, the contribution due to an impulse at zero frequency is removed by subtracting DELZ from each element of the characteristic function, $R(J)$. Second, after entering a DO loop on X, the result from Step 1 is inverse fourier transformed using utility subroutine IFT. IFT produces a value for $f(X)$ for each passage through the loop. In addition, X is incremented by DELX on each passage. The values of $f(X)$ and X are stored in $FX(1,J)$ and $FX(2,J)$, respectively. The Index J runs from 1 to NX.

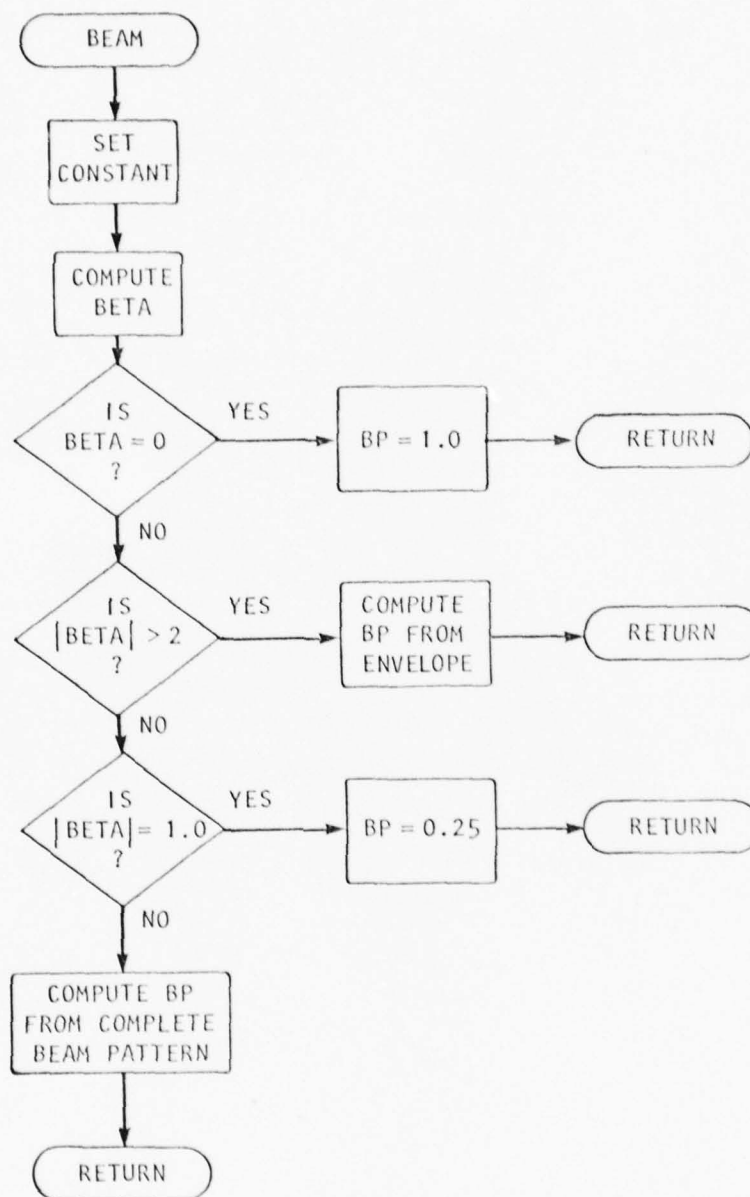


FIGURE 13 Flowchart for SUBROUTINE BEAM.

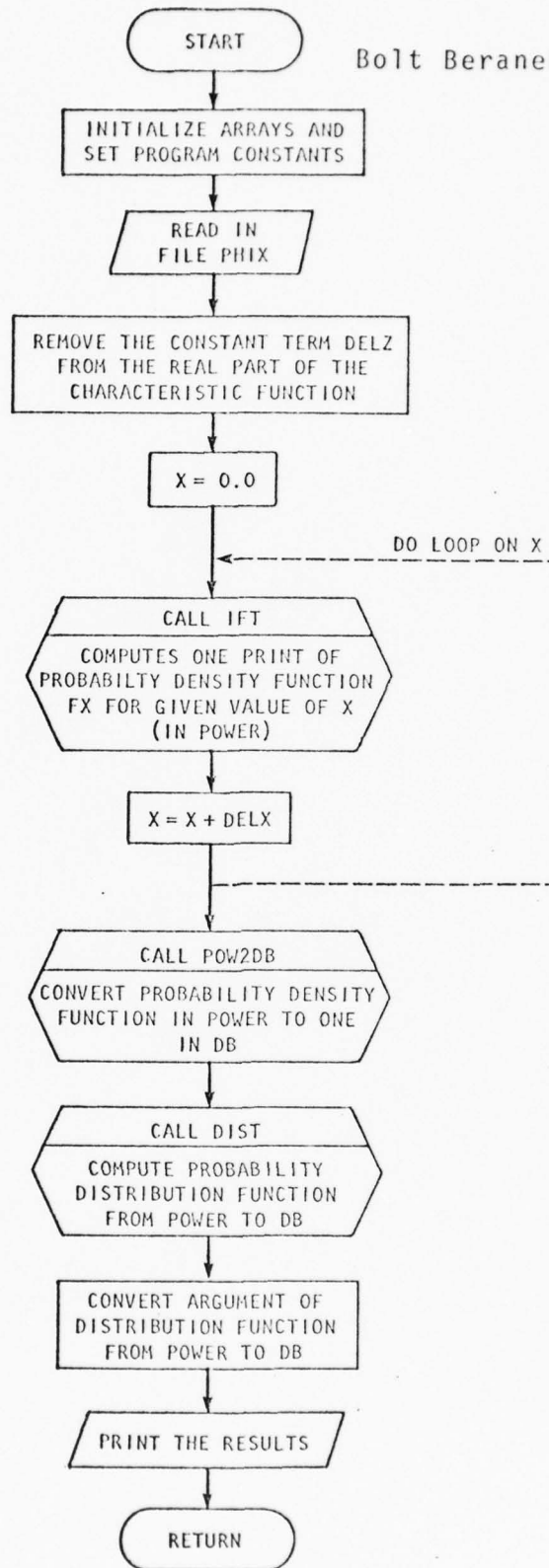


FIGURE 14 Program DENS Flowchart.

Report 3654

Bolt Beranek and Newman Inc.

The second function that is calculated by DENS is the probability density in db (re/watt) of the noise power at the beamformer output. It is found by using the utility subroutine POW2DB to convert the density function in power, $f(X)$, into a density function in db, $f(y)$. However, the point $X = 0$ watt is skipped; therefore, $f(y)$ has one less sample point than $f(X)$. The values of $f(y)$ at the sample points y are stored in $FY(1,J)$, while the sample points are stored in $FY(2,J)$.

The third function that is calculated by DENS is the distribution function (in db re/watt) for noise power at the beamformer output. This function is found by using the utility subroutine DIST, which produces a distribution function in power from a density function in power. The distribution function (in db) is calculated in two steps; first, DIST is called to operate on FX . This produces $DF(I,J)$, where the $DF(1,J)$ are the sampled distribution function (in power), and the $DF(2,J)$ are the sample points S_j . Second, the abscissa of the distribution, $DF(2,J)$ are converted to db re/watt, and stored back in $DF(2,J)$ (note the point $X = 0$ is skipped).

Output data are printed at the terminal. This includes descriptive information (such as center frequency, band width, beam steer angle, etc.) that is pertinent to the case at hand, followed by three tables of the density and distribution functions.

The maximum number of sample points is fixed in the program as 301 for R and for FX (i.e., $NX = 301$). While the maximum number for FY , and DF are one less, 300. The sample point spacing of $f(X)$, $DELX$ is equal to 10^{-4} . These limitations can be changed easily by altering COMMON, DIMENSION, and DATA statements.

4.3 Output

The output from Program DENS is printed at the computer terminal. This is the final output of the Ambient Noise Model for the particular case of interest.

The program prints the identification number for the particular route and sensor geometry; the frequency band and the center frequency; the beam steering angle and beam width of the main lobe of the array; the total number of ships; the probability of no noise; the mean, variance and

Report 3654

Bolt Beranek and Newman Inc.

standard deviation of beamformer output noise power in units of watts; the probability density function of noise power in watts; the probability density function of noise power in db; and the probability distribution function of noise power in db. Table 6 shows the structure of the printed output from Program DENS.

4.4 Glossary and Program Listing

The following glossary contains definitions of the FORTRAN constants and variable names used in Program DENS. The names are presented in alphabetical order.

BSTER	- beam steer angle (deg.)
BWIDE	- width of major lobe (deg.)
DELF	- sample point spacing in characteristic function (radians/sec).
DELFQ	- beamformer bandwidth (Hz).
DELW1	- sample point spacing; used in subroutine IFT (radians/sec).
DELX	- sample point spacing of density function $f(X)$.
DELZ	- probability of no noise.
DF(1,J)	- distribution function of beamformer output noise at sample point J.
DF(1,J)	- sample point J of distribution (db/re watt).
FREQ	- beamformer center frequency (Hz).
FX(1,J)	- probability density (in power) of beamformer output noise at point X;
FX(2,J)	- sample point X; in probability density (watts)
FY(1,J)	- probability density (in db re/watt) of beamformer output noise at point y.
FY(2,J)	- sample point y; in probability density (db re/watt)

Report 3654

Bolt Beranek and Newman Inc.

TABLE 6 STRUCTURE OF THE OUTPUT OF
PROGRAM DENS

(NOTE: Each line represents one line of printout)

ID NUMBER = _____
 FREQUENCY = _____ HZ
 CENTER FREQUENCY = _____ HZ
 BEAM STEERING ANGLE = _____ DEG
 BEAM WIDTH (MAJOR LOBE) = _____ DEG
 TOTAL AVG. NUMBER OF SHIPS = _____
 PROBABILITY OF NO NOISE = _____
 MEAN BEAMFORMER OUTPUT NOISE POWER = _____ WATTS
 VARIANCE OF BEAMFORMER OUTPUT NOISE POWER =
 _____ WATTS**2
 STAND. DEV. IN OUTPUT NOISE POWER =
 _____ WATTS

	PROBABILITY DENSITY (FX)	•	POWER (WATTS) (X)
	•
NX LINES	{	•	•
		•	•
		•	•
		•	•
		•	•
	PROBABILITY DENSITY (FY)	•	POWER (DB) (Y) (RE/WATT)
	•
NX-1 LINES	{	•	•
		•	•
		•	•
		•	•
		•	•
	DISTRIBUTION FUNCTION (DF)	•	POWER (DB) (Y) (RE/WATT)
	•
NX-1 LINES	{	•	•
		•	•
		•	•
		•	•
		•	•

Report 3654

Bolt Beranek and Newman Inc.

IDNUM - identification number for this case.

J1M - number of sample points; used in subroutine IFT = NF.

JN - number of sample points -1; used internally to skip X = 0 in db conversions.

NF - number of sample points in characteristic function.

NX - number of sample points in f(X).

PHIX - input data file.

R(J) - real part of characteristic function; used in subroutine IFT.

TEMP(I,J) - working array.

TSMIP - total mean number of ships.

XMEAN - mean beamformer output noise power (watts).

XSTD - standard deviation of beamformer output noise power (watts).

XVAR - variance of beamformer output noise power (watts²).

The listing of Program DENS is as follows:

```

PROGRAM DENS(OUTPUT,PHIX,TAPE1=PHIX)
C-----
C THIS PROGRAM CALCULATES THREE FUNCTIONS THAT ARE USED IN THE
C AMBIENT NOISE MODEL . FIRST, IT FINDS THE PROBABILITY DENSITY
C FUNCTION ( IN POWER ) FROM ITS CHARACTERISTIC FUNCTION. SECOND,
C THIS DENSITY FUNCTION IN POWER IS CONVERTED INTO A DENSITY
C FUNCTION IN DB/ RE 1 WATT. THIRD, THE DISTRIBUTION FUNCTION
C ( IN DB ) IS CALCULATED. WRITTEN 21JULY1978 BY W.SCOTT.
C-----
COMMON/COM/R(301),DELW1,J1M
DIMENSION FX(2,301),FY(2,300),DF(2,301),TEMP(2,300)
DATA R/301*0.0/,NX/301/,DELX/1.0E04/,TEMP/600*0.0/
DATA FX/602*0.0/,FY/600*0.0/

REWIND 1

C READ IN TRASH
  READ(01,100)
100 FORMAT(1X,26H

```

Report 3654

Bolt Beranek and Newman Inc.

```

C READ IDNUM, FREQ, DELFQ
  READ(01,150) IDNUM, FREQ, DELFQ
150 FORMAT(1X,I3,2F10.4)
C READ BEAM STEERING ANGLE AND MAJOR LOBE WIDTH
  READ(01,200) BSTER, BWIDE
200 FORMAT(1X,2F15.4)
C READ TSHIP, DELZ, XMEAN, XVAR
  READ (01,250) TSHIP, DELZ, XMEAN, XVAR
250 FORMAT(1X,E11.4,3E20.14)
C READ NF, AND DELF
  READ(01,300) NF, DELF
300 FORMAT(1X,I4,E20.14)
C READ IN THE CHARACTERISTIC FUNCTION
  READ(01,350) ( R(K),K=1,NF)
350 FORMAT(1X,E20.14,/)
C SET THE NUMBER OF SAMPLE POINTS AND THE FREQUENCY SPACING
C IN THE CHACTERISTIC FUNCTION.

  JIM = NF
  DELW1 = DELF

C REMOVE CONTRIBUTION FROM DELTA FUNCTION AT FREQ=0.0 .
  DO 400 I = 1,NF
    R(I) = R(I)-DELF
  400 CONTINUE

C CALCULATE THE PROBABILITY DENSITY IN POWER

  X = 0.0
  DO 450 J = 1,NX
    CALL IFT(GX,X)
    FX(1,J) = GX
    FX(2,J) = X
    X = X+DELF
  450 CONTINUE

C FIND THE PROBABILITY DENSITY IN DB/RE 1WATT.(SKIP X=0.0)

  JN = NX-1
  DO 500 J = 1,JN
    TEMP(1,J) = FX(1,J+1)
    TEMP(2,J) = FX(2,J+1)
  500 CONTINUE

  CALL POW2DB(TEMP,JN,FY)

C FIND THE DISTRIBUTION FUNCTION IN DB/RE 1WATT.
C FIRST FIND THE DISTRIBUTION FUNCTION IN POWER; THEN,
C CONVERT THE ABSCISSAE TO DB.

```

Report 3654

Bolt Beranek and Newman Inc.

```

CALL DIST(FX,NX,DF,DELZ)
DO 550 K = 2,NX
    DF(2,K) = 10.*ALOG10(DF(2,K))
550 CONTINUE
XSTD = SQRT(XVAR)

```

C PRINT DATA AT TERMINAL

```

PRINT 1900
PRINT 650
650 FORMAT(1X,40H THE FOLLOWING DATA ARE THE RESULTS OF)
PRINT 700
700 FORMAT(1X,36H PROGRAM DENS, WRITTEN 21JULY78 BY)
PRINT 750
750 FORMAT(1X,11H W.SCOTT.)
PRINT 1900
PRINT 800, IDNUM
800 FORMAT(1X,12HID NUMBER = ,I3)
PRINT 850, FREQ
850 FORMAT(1X,12HFREQUENCY = ,F10.4,3H HZ)
PRINT 900, DELFC
900 FORMAT(1X,19HCENTER FREQUENCY = ,F10.4,3H HZ)
PRINT 950, BSTER
950 FORMAT(1X,22HBEAM STEERING ANGLE = ,F15.4,4H DEG)
PRINT 975, BWIDE
975 FORMAT(1X,19HMAJOR LOBE WIDTH = ,F15.4,4H DEG)
PRINT 1000, TSHIP
1000 FORMAT(1X,29HTOTAL AVG. NUMBER OF SHIPS = ,E11.4)
PRINT 1050, DELZ
1050 FORMAT(1X,26HPROBABILITY OF NO NOISE = ,E12.6)
PRINT 1100, XMEAN
1100 FORMAT(1X,37HMEAN BEAMFORMER OUTPUT NOISE POWER = ,
1 E12.6,6H WATTS)
PRINT 1150
1150 FORMAT(1X,44HVARIANCE OF BEAMFORMER OUTPUT NOISE POWER = )
PRINT 1175, XVAR
1175 FORMAT(1X,E12.6,9H WATTS**2)
PRINT 1200
1200 FORMAT(1X,35HSTAND. DEV. IN OUTPUT NOISE POWER =)
PRINT 1225, XSTD
1225 FORMAT(1X,E12.6,6H WATTS,////)
PRINT 1250
1250 FORMAT(1X,27H PROBABILITY . POWER(WATTS))
PRINT 1300
1300 FORMAT(1X,22H DENSITY . (X))
PRINT 1350
1350 FORMAT(1X,14H (FX) .)
PRINT 1400
1400 FORMAT(1X,27H.....)
PRINT 1500, (FX(1,J),FX(2,J),J=1,NX)

```


Report 3654

Bolt Beranek and Newman Inc.

```
1500 FORMAT(1X,E12.6,3H . ,E12.6)
      PRINT 1900
      PRINT 1550
1550 FORMAT(1X,27H PROBABILITY . POWER(DB) )
      PRINT 1600
1600 FORMAT(1X,22H DENSITY . (Y))
      PRINT 1650
1650 FORMAT(1X,25H (FY) . (RE/WATT))
      PRINT 1400
      PRINT 1700, (FY(1,J),FY(2,J),J=1,JN)
1700 FORMAT(1X,E12.6,3H . ,F7.2)
      PRINT 1900
      PRINT 1750
1750 FORMAT(1X,25H DISTRIBUTION. POWER(DB))
      PRINT 1800
1800 FORMAT(1X,22H FUNCTION . (Y))
      PRINT 1850
1850 FORMAT(1X,25H (DF) . (RE/WATT))
      PRINT 1400
      PRINT 1700, (DF(1,J),DF(2,J),J=2,NX)
      PRINT 1900
1900 FORMAT(1X,///)
      REWIND 1
      STOP
      END
```

Report 3654

Bolt Beranek and Newman Inc.

4.5 Subroutines

This section describes the three subroutines that are called from the main program of DENS.

4.5.1 Subroutine IFT

In Section 3.1 of BBN Report 3653, a numerical method was developed to evaluate the special form of the inverse Fourier transform given by:

$$f(x) = \frac{2}{\pi} \int_0^{+\infty} R(\omega) \cos \omega x \, d\omega,$$

for $x > 0$, where $R(\omega)$ is the real part of the characteristic function of the probability density function $f(X)$ of the positive real random variable X .

The purpose of this section is to document the Subroutine IFT, which implements this numerical method.

The calling sequence is Subroutine IFT(FX,X) where X is the value of the average square pressure and FX is the value of the probability density function returned by the subroutine. Data is passed from the main program to the subroutine in a common storage area labeled COM, which contains the following:

- R(301) - an array containing the real part of the characteristic function,
- DELW1 - the integration step size in the w domain,
- J1M - the number of sample points in the w domain.

Figure 15 presents a flowchart for SUBROUTINE IFT. Upon entry, the subroutine tests to see if X is equal to zero. If X is equal to zero then $f(0)$ is computed by a numerical approximation to the equation

$$f(0) = \frac{2}{\pi} \int_0^X R(\omega) \, d\omega$$

Report 3654

Bolt Beranek and Newman Inc.

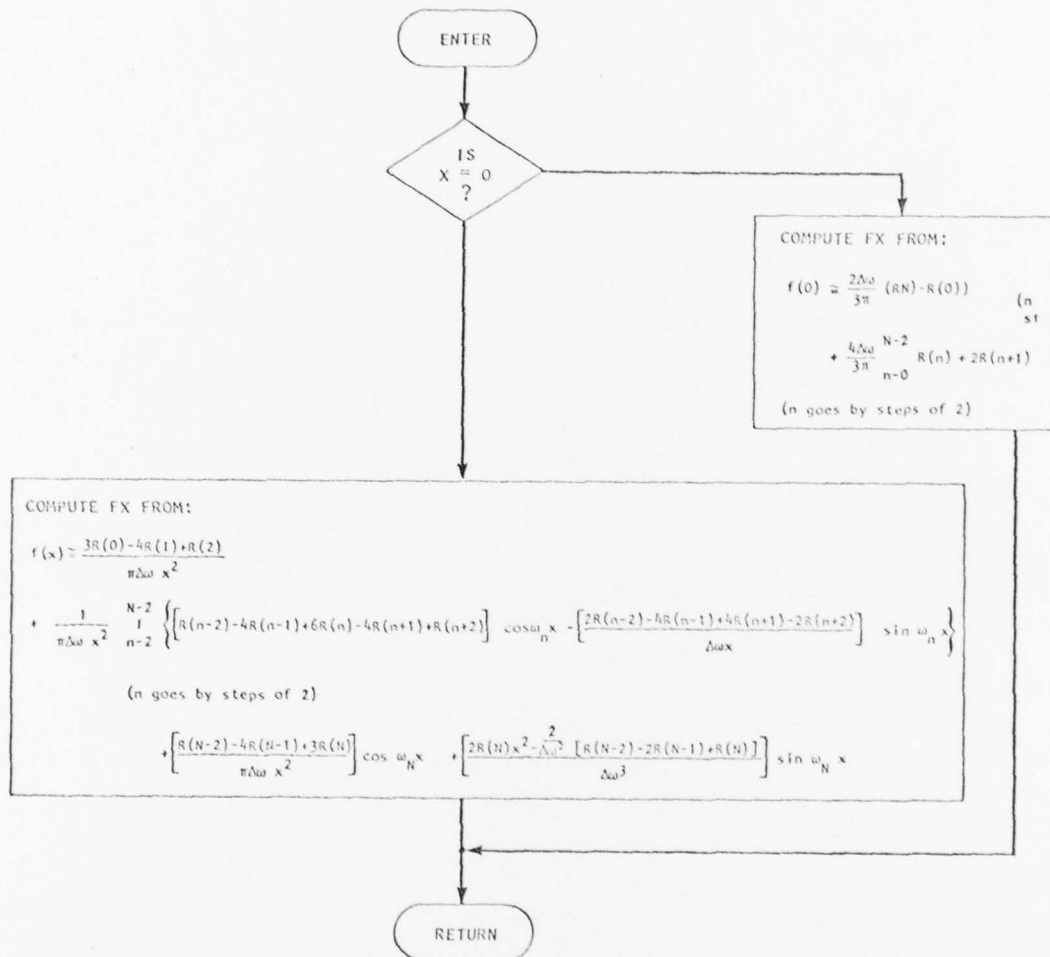


FIGURE 15 Flowchart of Subroutine IFT.

Report 3654

Bolt Beranek and Newman Inc.

developed in Section 3.1 of BBN Report 3653. If $X > 0$, then $f(X)$ is computed by the appropriate formula developed in Section 3.1 of BBN Report 3653.

A listing of Subroutine IFT is as follows;

```

C      SUBROUTINE IFT(FX,X)
      COMPUTES INVERSE FOURIER TRANSFORM
      COMMON/COM/R(301),DELW1,J1M
      PI=3.14159265
      K1=J1M-2
      IF(X.NE.0.0)GO TO 150
      FACT1=2.0/(3.*PI)
      SUM1=0.
      DO 100 K=1,K1,2
      SUM1=SUM1+R(K)+2.*R(K+1)
100    CONTINUE
      SUM1=DELW1*(R(J1M)-R(1)+2.*SUM1)
      FX=FACT1*SUM1
      GO TO 300

C
150    C=COS(2.*DELW1*X)
      S=SIN(2.*DELW1*X)
      Y2=1.
      Y1=0.
      SUM1=0.
      FACT2=1./(PI*X*X)
      FACT3=FACT2/X
      DO 200 K=3,K1,2
      FACT4=R(K-2)-4.*R(K-1)+6.*R(K)-4.*R(K+1)+R(K+2)
      FACT5=(2.*R(K-2)-4.*R(K-1)+4.*R(K+1)-2.*R(K+2))/(DELW1*X)
      SN=Y1*C+Y2*S
      CN=Y2*C-Y1*S
      Y1=SN
      Y2=CN
      SUM1=SUM1+FACT4*CN-FACT5*SN
200    CONTINUE
C
      SN=Y1*C+Y2*S
      CN=Y2*C-Y1*S
      Y1=SN
      Y2=CN
      SUM1=SUM1+3.*R(1)-4.*R(2)+R(3)
      SUM1=SUM1+(R(J1M-2)-4.*R(J1M-1)+3.*R(J1M))*CN
      SUM1=SUM1*FACT2/DELW1
      TEMP=R(J1M-2)-2.*R(J1M-1)+R(J1M)
      TEMP=2.*(R(J1M)*X*X-TEMP/(DELW1*DELW1))
      FX=SUM1+TEMP*SN*FACT3

C
300    RETURN
      END

```

4.5.2 Subroutine POW2DB

This subroutine converts a probability density function of a random variable in power to a probability density function of a random variable in db.

Let Y be a random variable in units of db with probability density function $P_Y(y)$. Let X be a random variable in units of watts with probability density function $F_X(x)$. These random variables are related through the nonlinear transformation:

$$Y = 10 \log_{10} X \quad (1)$$

The inverse transformation is

$$X = 10^{Y/10} \quad (2)$$

It is assumed that $x \geq 0$.

Since probability measure is conserved, the following differential relationship is valid:

$$P_Y(y) dy = F_X(x) dx \quad (3)$$

From Equation 3 and the differential of Equation 2, the resulting formula for $P_Y(y)$, given $F_X(x)$ is:

$$P_Y(y) = (0.230258) 10^{y/10} F_X(10^{y/10}) \quad (4)$$

This subroutine implements Equations 1 and 4. The calling sequence for the subroutine is POW2DB (PX,N,PY) where N is the number of sample points in X and in Y ; PX is a $(2 \times N)$ dimensioned array containing the density where

$$PX(1,J) = F_X(x) \text{ and}$$

$$PX(2,J) = X; \text{ and}$$

PY is a (2xN) array containing the computed values of the density function in db where $PY(1,J) = P_Y(y)$ and $PY(2,J) = Y$. Figure 16 shows the flowchart for this subroutine. A listing of the subroutine is as follows:

```
      SUBROUTINE POW2DB(PX,N,PY)
      DIMENSION PX(2,N),PY(2,N)
C
C SUBROUTINE WRITTEN 30 NOV 1976 BY ZESKIND
C CONVERTS PROBABILITY DENSITY FUNCTION FROM POWER RATIO
C TO DB RE: MICRO PASCAL
C
C N= NUMBER OF SAMPLE POINTS
C PX(1,J)= P(X)= DENSITY IN POWER
C PX(2,J)= X = POWER RATIO
C PY(1,J)= P(Y)= DENSITY IN DB
C PY(2,J)= Y = 10 LOG X = DB
C
      A=(10.0)**(0.1)
      DO 100 J=1,N
      PY(2,J)=10.0*ALOG10(PX(2,J))
      PY(1,J)=(0.23025850929940)*(A**PY(2,J))*PX(1,J)
100  CONTINUE
      RETURN
      END
```

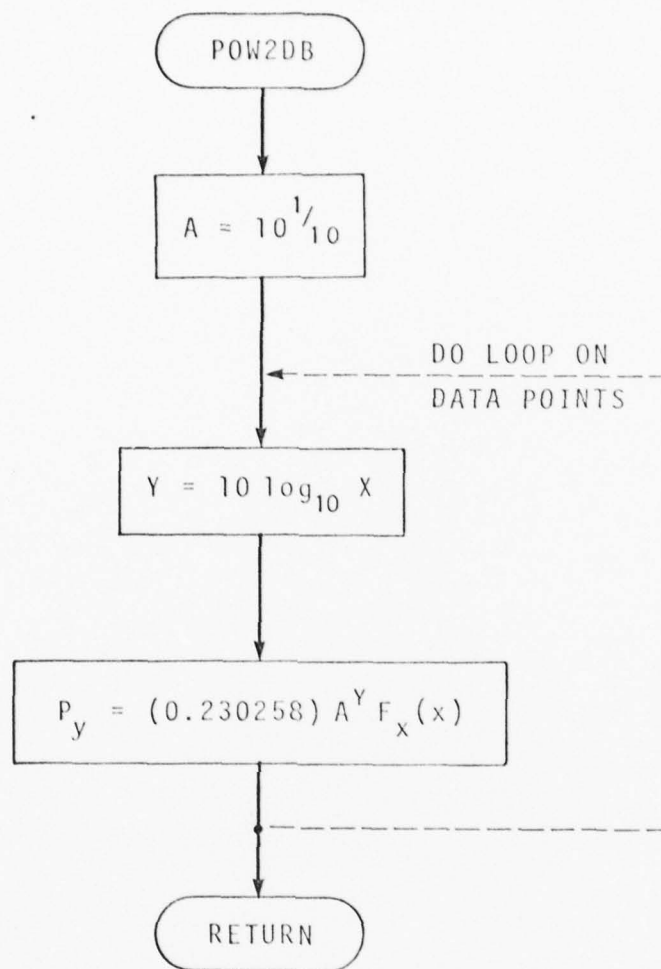



FIGURE 16 Flowchart of SUBROUTINE POW2DB.

4.5.3 Subroutine DIST

This subroutine computes the probability distribution function $D_X(x)$ from the input density function $F_X(x)$. It is assumed that $F_X(x)$ only has non-zero values for $x \geq 0$. It is further assumed that $F_X(x)$ can have a unit impulse function $\delta(x)$ at $x=0$ of magnitude A . The formula relating D_X to F_X is given by:

$$D_X(x) = A \delta(x) + \int_0^x F_X(\xi) d\xi$$

Subroutine DIST implements this equation. The trapezoidal method of integration is used to approximate the integral, with an integration step size depending on the sample points of $F_X(x)$.

The calling sequence for the subroutine is DIST (F,N,DF, DELTA) where N is the number of sample points; F is a $(2 \times N)$ dimensioned array containing the probability density function $F_X(x)$ in row $F(1,K)$ and the corresponding value of x in row $F(2,K)$; DELTA is the magnitude of the unit impulse function; and DF is a $(2,N)$ dimensioned array containing the computed distribution function $D_X(x)$ in row $DF(1,K)$ and the corresponding value of X in row $DF(2,K)$.

Figure 17 shows the flowchart for Subroutine DIST. The variable S is a running sum, initialized to DELTA, which represents the integral as the DO loop steps through the values of X . DELX is a variable which is equal to one-half the integration step size at each value of X . A listing of the subroutine is as follows:

```

SUBROUTINE DIST(F,N,DF,DELTA)
  DIMENSION F(2,N),DF(2,N)
  C
  C WRITTEN 2 DEC 1976 BY ZESKIND
  C CALCULATES THE DISTRIBUTION FUNCTION FROM THE INPUT
  C DENSITY FUNCTION. DENSITY FUNCTION HAS N SAMPLE POINTS.
  C TRAPEZOIDAL INTEGRATION .

```

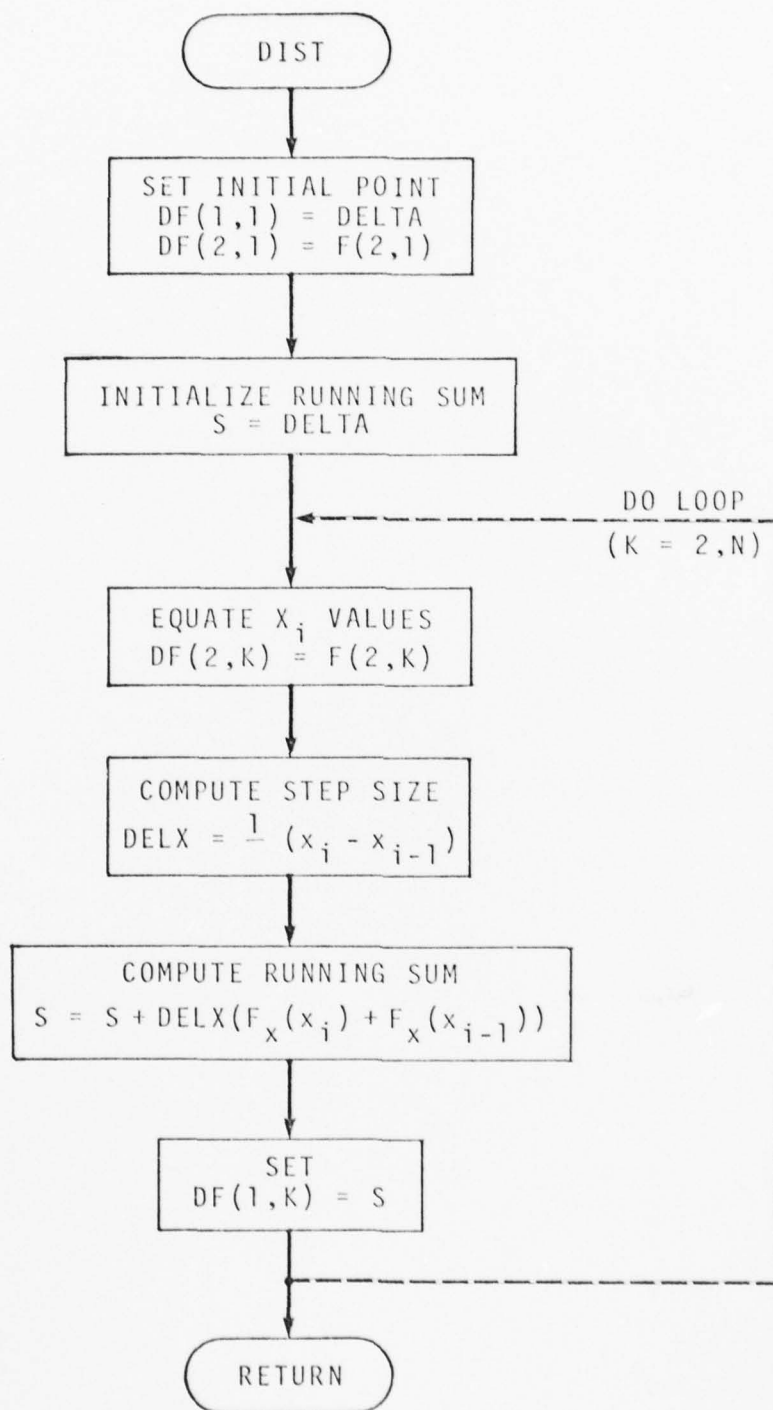


Figure 17 Flowchart of Subroutine Dist.

```
C DELTA= MAGNITUDE OF DELTA FUNCTION AT ORIGIN OF F(X).
C F(1,N)= DENSITY
C F(2,N)= X = POWER RATIO
C DF(1,N)= DISTRIBUTION FUNCTION
C DF(2,N)= X
C
      DF(1,1)=DELTA
      DF(2,1)=F(2,1)
      S=DELTA
      DO 40 K=2,N
      DF(2,K)=F(2,K)
      DELX=(F(2,K)-F(2,K-1))/2.0
      S=S+DELX*(F(1,K)+F(1,K-1))
      DF(1,K)=S
40    CONTINUE
      RETURN
      END
```